Security Analysis of Peer-to-Peer based Soft System Bus based Systems

Muhammad Anwarul Azim, Yuichi Goto, and Jingde Cheng

Department of Information and Computer Sciences,
Saitama University, Saitama, 338-8570 Japan

Summary
Asynchronous message queuing middleware is a dependable solution for reliable communication in distributed computing environment. Soft System Bus (SSB) is an asynchronous messaging middleware that facilitates availability, reliability and continuity to SSB based Systems (SSBBSs). An SSBBS can be maintained, upgraded and reconfigured during run time without stopping its reactions even when it had some trouble or it is being attacked. A peer-to-peer (P2P) based implementation of an SSBBS was designed for large scale enterprise. However, the security analysis was not sufficient for P2P based SSBBSs. This paper gives a security analysis for P2P-based SSBBSs, and lists up security threats including new types of security threats for P2P-based SSBBSs. The paper also shows security requirements and technical issues for P2P-based SSBBSs.

Key words:
Security analysis, Peer-to-Peer, Asynchronous middleware, Persistent Computing System, Security threat

1. Introduction

Many commercial/noncommercial middlewares are gaining more attention in large enterprises for building highly available asynchronous messaging systems and for integrating heterogeneous distributed applications. Soft System Bus (SSB) emerged to meet the need of middleware that has the functionality of persistent computing system (PCS), a paradigm that aims to develop continuously dependable and dynamically adaptive reactive-systems [1]. A Soft System Bus based System (SSBBS) is a system comprised of SSB, some functional components and control components. Brilliant features such as availability, reliability, continuity, asynchronous and dynamic broker network, application's independence from single node/site, less administrative overhead and low deployment cost of SSB make it important. Peer-to-Peer (P2P) based SSBBs emerged to meet the implementation need of middleware that has the functionalities.

By no means, security is a crucial fact for any system. Existing attackers in the real world will target the SSBBS to find out vulnerabilities and try to exploit. When an adversary is motivated and capable of exploiting a specific vulnerability, then the system is in danger [6], [7]. Up to now, no one has investigated security threats to SSBBS in detail. A work has been done before head by T. Endo et. al. on security of PCS which defined the basic security requirements and functions in persistently reactive systems [2]. They proposed the framework of SSB-connectors with specifications. Above work is not sufficient for the security of SSBBS as it was published before distributed peer-to-peer based design and implementation of SSB. And in this paper our context in P2P based SSBBS. We will focus on P2P based SSBBS in this paper. So, in this paper P2P based SSBBS is alternatively mentioned as SSBBS.

There are several reasons for not addressing the security concern for the SSBBS during earlier design and implementation. The purpose of the design and implementation included design and evaluation of SSB for large-scale PCSs by eliciting and analyzing the requirements. And ultimate purpose was to build the general purpose part of SSBBSs, i.e., the SSB and Control Components (CCs) which is collectively called SSB package that will be used to build large-scale long lasting reactive systems. Functional Components (FCs) will be designed, developed and added from application area. Because of that, security was not taken within the scope of earlier work.

Security analysis is important for building systems to remain dependable in the face of malice, error, or mischance. This paper gives a security analysis for P2P-based SSBBSs, lists up security threats including new types of security threats for P2P-based SSBBSs, and shows security requirements and technical issues for SSBBSs.

The remainder of this paper is structured as follows. Section 2 describes the SSBBS. Section 3 gives the security analysis. Technical issues are illustrated in section 4. And section 5 concludes the paper.

2. SSBBS

2.1 Architecture

Manuscript received February 5, 2011
Manuscript revised February 20, 2011

An SSB is simply a communication channel with the facilities of data/instruction transmission and preservation to connect components in a component-based system. It has some data/instruction stations, which have the facility of data/instruction preservation, connected sequentially by transmission channels, both of which are implemented in software techniques, such that over the channels data/instructions can flow among data-instruction stations and a component tapping to a data-instruction station can send data/instructions to and receive data/instructions from the data-instruction station. SSB has connection with all components and provides hardware and platform independent middleware support to the components. Data/Instruction Stations (DISs) are considered as nodes of SSB with communication and data and instruction preservation facility. Any two components are not allowed to communicate directly but via network of DISs.

An SSBBS consists of a number of components and one or more SSBs that functions continuously anytime without stopping its reactions even when it is being maintained, upgraded, or reconfigured, it had some trouble, or it is being attacked. There are two types of components in an SSBBS: Control Components (CCs) and Functional Components (FCs). CCs are for a general-purpose, and measure, monitor and control the legitimate operation of communication. CCs also control the FCs. FCs are for application specific, and are developed by the application developers and provides functionality to the applications. Figure 1 shows an SSBBS.

In a structured P2P based implementation of SSBBS, P2P network of DISs works as reliable point-to-point communication channel among applications, whereas, the traditional p2p network is used either as distributed storage

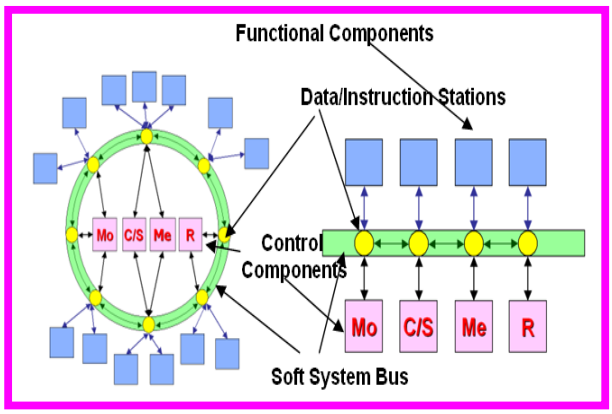

Figure 1: An SSBBS

or as middlewares for publish-subscribe systems. In that implementation, a distributed hash table protocol based on Chord for large scale peer to peer systems was designed [3],[5]. The authors focused on the mechanisms to solve reliability and availability issues and analyze the effect of failures/arrivals of nodes on availability [4], [8].

In SSBBSs, DISs are comprised of software abstraction layers (replication layer, routing layer and network layer), relationships among the layers and externally visible interfaces that are used by other DIS stations or components.

DISs ensure replicating the message in the neighboring DISs, if it cannot be delivered immediately to the destination component and when a DIS fails, its neighbor takes over its responsibility.

Application field of SSBBS includes distributed enterprise like FedEx, reliable messaging system among large number of employees of a big enterprise, large-scale long lasting reactive systems, e.g., air traffic control systems, operating systems, transportation/industrial control systems, etc.

2.2 Replication Mechanism and Message Protocol

Three types of data are replicated in the replication layer of DIS: a) The messages that cannot be delivered to the components because of the unavailability of the components, b) The profile (id, IP address, public key and credentials) of components and DISs and c) type of the component (control or functional) and its privileges.

Each DIS replicates its states to some other DISs. They share resources cooperatively. If one DIS fails, neighbor DIS, containing the replica, takes over its responsibility and a new replica is created dynamically.

Source and destination component communicate via source, destination and/or intermediate DIS with the acknowledgment and message preservation strategy.

Message is consumed by an application reliably, in-order and exactly once even in case of broker failures and faults to provide availability and continuity.

3. Security Analysis

3.1 Participants and Information Assets

This is very important to list up the participants in the operations of SSBBS to have clear scenario of the information leak hole while the system is under threat or suspect of threat. Many participants, involved in various operations in a SSBBS are listed up to start the security analysis of SSBBS.

- System admin: The system admin will be responsible for working the system continuously when it is maintained,

updated, reconfigured or under attack. Can create all types of account applicable. Can check all operations. Can check other admin and user activities.

- SSB server admin: The system will be spread over several geographical locations. Each location will be facilitated by a SSB server admin.
- SSB monitor: SSB monitor will responsible for monitoring the system operations so that any illegal activity can be stopped. Admin can also do this.
- SSB analyst: The SSB analyst will look after the system report to analyze several facts so that improvements or desired quality can be ensured.
- SSB developer: SSB developer will write the program code in any purpose.
- Application server admin: The application server admin will take care of operation and performance of specific application for example: web server.
- Application Monitor: Application monitor will responsible for monitoring the system operations so that any illegal activity can be stopped. Admin can also do this.
- Application analyst: The application analyst will look after the system report to analyze several facts so that improvements or desired quality can be ensured.
- Application developer: The application programming is the task of application developer.
- Terminal admin: The terminal admin the technical person who is installing, maintaining the software and hardware to use the application properly.
- Application user: Application user is the end user who is interacting with the application.
- Attacker: Attacker will be performing various attacks.

The relationship among the participants is important. Because it gives us understanding of the interactions within the system. Figure 2 shows the relationship among the participants.

Data and information are the lifeblood. Data is recognized as a vital enterprise asset in the Information Age. The capture and misuse of such assets cause loss of control, and harm investments and enterprise objectives. For this reason, investigation of information asset of SSBBS is necessary. Here we show who creates which information assets in an SSBBS.

- System admin : System admin account, System admin id, Update-history, Maintain-history, Reconfig-history, SSB procedure, SSB policy, Security policy, measured data, Control/schedule data, and record data.
- SSB server admin: SSB Server Admin account, SSB server admin id, backup message, data/instruction to replicate, order list of message, Application setup history, SSB source code.
- SSB monitor: SSB monitor account, SSB monitor id, Replication layer log, Routing layer log, Measuring log, Control and synchronization log, Recording log, Monitoring log, SSB error log, SSB access log, Security log.
- SSB analyst: SSB analyst account, SSB analyst id, System report.
- SSB developer: SSB Source code.

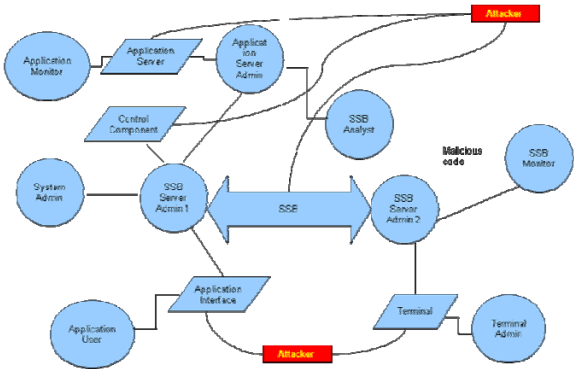

Figure 2: The relationship among the participants

- Application server admin: Application server account, Application server id, Application procedure, Application policy, Application user (e.g. customer) profile, Application source code.
- Application Monitor: Application monitor account, Application monitor id, Application log, Application user personal log, Application error log, Application access log.
- Application analyst: Application report.
- Application developer: Application source code.
- Terminal admin: Terminal access log, terminal error log, terminal source code.
- Application user: User signature.
- Attacker: Any malicious info.

It is important to analyze the correct permissions granted to participants of SSBBS to operate on the information asset. Because, this analysis is necessary to work with repudiation threat. We, therefore, analyzed who can create, read, use, update, and delete which data in SSBBS. Table 1 shows permissions to information assets for participants to create, read, update and delete.

Table 1: Permissions to information assets for participants to create, read, update and delete. In the table, c, r, (r), u, d mean create, read, read and use, update, and delete respectively.

Assets	System admin	SSB server admin	SSB monitor	SSB analyst	SSB developer	Application server admin	Application Monitor	Application analyst	Application developer	terminal admin	Application user
Representation											
Main-memory database of SSB	(r), u					-	-	-	-	-	-
TCP packet	r					-	-	-	-	-	-
IP packet	r					-	-	-	-	-	-
Inter Process Communication Message	r					-	-	-	-	-	-
Remote Procedure Call message	r					-	-	-	-	-	-
Message queue in SSB	r					-	-	-	-	-	-
interface between DIS and CC	(r)					-	-	-	-	-	-
Interface between DIS and FC	(r)					r	r	(r)	(r)	(r)	-
Information asset											
Application setup history	r	r	r	(r)		r	-	-	-	-	-
SSB Update-history	r, u	r	r			-	-	-	-	-	-
SSB Maintain-history						-	-	-	-	-	-
SSB Reconfig-history						-	-	-	-	-	-
SSB procedure						-	-	-	-	-	-
SSB policy						-	-	-	-	-	-
Security policy						?	-	-	-	-	-
Measured data						-	-	-	-	-	-
schedule/control data						-	-	-	-	-	-
Record data						-	-	-	-	-	-
order list of message	r, u					-	-	-	-	-	-
backup message	r					-	-	-	-	-	-
data/instruction to replicate						-	-	-	-	-	-
Application log in SSB						-	-	-	-	-	-
Replication layer log						-	-	-	-	-	-
Routing layer log						-	-	-	-	-	-
Measuring log						-	-	-	-	-	-
Control and synchronization log						-	-	-	-	-	-
Recording log						-	-	-	-	-	-
Monitoring log						-	-	-	-	-	-
SSB access log						-	-	-	-	-	-
SSB server error log						-	-	-	-	-	-
Security log						-	-	-	-	-	-
System report						-	-	-	-	-	-
SSB source code	r,d	r, d	-	-	c, r, u, d	-	-	-	-	-	-

System admin account	c, r, u, d					-	-	-	-	-	-
System admin id	c, r, u, d					-	-	-	-	-	-
SSB server admin account						-	-	-	-	-	-
SSB server admin id						-	-	-	-	-	-
SSB monitor account	c					-	-	-	-	-	-
SSB monitor id	c, r					-	-	-	-	-	-
System analyst account	c, d					-	-	-	-	-	-
System analyst id	c,r,d					-	-	-	-	-	-
Application Update-history	-	-	-	-	-						
Application Maintain-history	-	-	-	-	-						
Application Reconfig-history	-	-	-	-	-						
Application procedure	-	-	-	-	-						
Application policy	-	-	-	-	-						
Application user personal log	-	-	-	-	-						
Application user (e.g. customer) profile	-	-	-	-	-	(r),d	-	-	r	r	c, u,(r), d
Application Source code	-	-	-	-	-	r, d	-	-	c, r, u, d	-	-
Application access log	-	-	-	-	-						
Application server error log	-	-	-	-	-						
Application report	-	-	-	-	-						
Application Server admin account	-	-	-	-	-						
Application Server admin id	-	-	-	-	-						
Application monitor account	-	-	-	-	-						
Application monitor id	-	-	-	-	-						
Application analyst account	-	-	-	-	-						
Application analyst id	-	-	-	-	-						
Application user account	-	-	-	-	-						
Application user id	-	-	-	-	-						
Terminal admin account	-	-	-	-	-	c, r, u, d	-	-	-	r, u	-
Terminal admin id	-	-	-	-	-	c, r, u, d	-	-	-	r, u	-
Terminal access log	-	-	-	-	-	-	-	-	-	r, d	-
Terminal error log	-	-	-	-	-	-	-	-	-	r, d	-
Application user signature	-	-	-	-	-	-	-	-	-	-	c, (r), u, d

3.2 Threat Analysis

In this paper, we discussed about the familiar attacks in distributed systems that are also possible to SSBBS and we showed some security threats that are special to SSBBS.

Malicious code injection: This attack can easily inject malicious code to the SSB source code or application source code section of the target. This attack can be done by SSB developer or Application developer because they have access. In this case, it is an insider attack which is huge troublesome to tackle. Malicious code injection can also be done by attackers.
Man in the middle attack: The attacker can intercept the traffic between any two entities, e.g. two DISs. Then attacker can pretend as one of them. It may have the form of application user or (system/server/application) admin and thereby can harm - routing table, replicated data, replicated instruction, and profile.
Denial of service (DoS) attack: This attack attempts to make a computer resource unavailable to its intended users. It can harm routing packet, group information and interface, replicated data, replicated instruction, profile, libraries, measured information, and monitored information. There are many form of DoS attack. Smurf, Fraggle, Ping Flood, SYN flood, Land, Teardrop, Bonk and Boink are well known DoS attacks.
Spoofing attack: The attacker will pretend to send or receive the information asset look like it was from someone else.
IP address spoofing, teardrop, pharming attack by adversary can harm IP packets.
The mathematical & brute force attacks are supposed to be most serious attack on SSBBS. These types of attacks can harm enormous information assets.

Any architecture that relies on multiple independent or semi-independent components can be susceptible to resource deadlocks or conflict. Conflict between control components & functional components in SSBBS can be done with the aim of information stealing, denial of service, information manipulation by flipping any specific operation and/or information in the SSBBS protocol for example changing a flag value. Conflicting attack can be done by developers by varying components.

P2P technique (Chord/Pastry) and SSB technique that are integrated in SSBBS can be conflicted with each other or can be saturated/disjointed by the attacker.

Component forgery inside p2p:
Because of SSBBS architecture, the typical identity theft will become critical by component forgery by the attacker.

Categorization of threats:
There are similarities among the threats on the basis of loss they cause to the system. Based on the similarity we categorize the threats into 4 main types.
a) Interrupt: An asset of the SSBBS is destroyed or becomes unavailable or unusable. The following threats are in this category: Denial of Service, Man in the middle, conflict between functional and control components, conflict between SSB & P2P technique.
b) Interception: An unauthorized party (a person, a program, or a computer) gains access to an information asset of SSBBS. The following attacks are in this category: Man in the middle, spoofing attack, malicious code injection.
c) Modification: An unauthorized party not only gains access to but tampers with an information asset of SSBBS. The following attacks are in this category: Man in the middle, spoofing attack, malicious code injection.
d) Fabrication: An unauthorized party inserts counterfeit objects into the system. The following attacks are in this category: Man in the middle, Brute force attacks, Mathematical attack.

3.3 Security Requirements

We outline the requirements for building a secure SSBBS in this sub section.

R1: Availability: The system must not be unavailable due to security attacks.

R2: Confidentiality: Message transmitted must not be readable to unintended entities.

R3: Integrity: Message transmitted over network must be identical to the original.

R4: Authentication: Every administrator or user must be verified according to their claimed identity.

R5: Access Control: There must be mechanism of granting suitable rights to legitimate users.

R6: Accountability: User actions that are security-critical must be traceable.

R7: Prevent all attack: There are many kinds of attack to SSBBS. It is necessary to prevent all attacks to keep the system safe.

R8: Detect when attacked: Whenever there is an attack to the SSBBS, it must be detected.

R9: Working when attack handled: The SSBBS should not stop when the attack is being treated.

R10: Security enforcement efficiency: The attack handling technique should not be complex and time consuming.

R11: Cost effective: The security solution should not be expensive.

R12: Scalable: There should not be any constrain that the security solution only works for limited number of node. The solution should work for both small and big size network.

R13: Reconfigurable security: The security solution should adopt the environment change due to system update, component added, or component deleted.

R14: Resilience: The design of SSBBS has the resilience aim as it is important in P2P based systems where nodes are very dynamic in nature. So, security architecture should deliver the promised level of security assurances even if its composition changes over the time.

R14: Protection of layers: Breaches between layers (application-P2P routing, routing-network, network-transport) must be protected.

R15: Anti-Cloning: The SSBBS should be in defended such a way that the attacker with huge resource cannot make an exact clone of the system.

R16: Privacy: The anonymity is provided by the SSBBS when ii is needed.

3.4 Correspondences Between Threats and Requirements

In this sub section we place the correspondence between the threats and the requirements. As the threats to SSBBS are categorized into 4 groups, we write down their corresponding security requirements to protect.
a) Interruption: R1-Availability. The SSBBS should not be in dead state to provide availability.
b) Interception: R2-Confidentiality. Only the right person or entity should see the content or asset to maintain confidentiality.
c) Modification: R3-Integrity. The information asset cannot be modified without proper authority to ensure integrity.
d) Fabrication: R4-Authenticity. Fabrication type of threats can be protected if only the party with appropriate privilege can insert content to the system. Others are denied on the base of authenticity lacking.

Other security requirements are associated for security functionalities and other matters.

3.5 Candidate Solutions and Recommendations

The aim of this sub section is to illustrate the possible solutions existing at present for security requirements of SSBBS. We have to keep in mind that solutions to all the requirements are not possible to find. Some requirements have trade-off relationship with each other. For example, cost may have to be compromise to gain scalable and reconfigurable security.

Integrity- Message authentication codes (MACs), digital signatures can be used to ensure the integrity of data.

Confidentiality: The cryptography is a solution to maintain the confidentiality.

Authentication: Candidate solutions include PGP keys, SSH keys, SPKI, Secure HTTP.

Access Control: Mandatory access control (MAC), Discretionary access control (DAC) and Role based access control (RBAC). We recommend RBAC for SSBBS. Because MAC and DAC are not suitable for SSBBS.

3.6 A Proposed SSBBS Design Modification

We propose to add one control component in the SSBBS which will be responsible for enforcing the security and handle all security issues. This security control component will have a local firewall apart from the operating system firewall. It will protect an SSBBS from malicious outsiders. Secure control component is responsible to keep audit log and other security measurements.

4. Technical Issues

In this section we illustrate the technical issues related to SSBBS security.

The ISO/IEC 27002 has Security Policy standards that can be a reference for security policy determination of SSBBS. This is a good start point for the security policy developer of SSBBS. But fitting the total SSBBS appropriately with ISO/IEC 27002 Security Policy standards is an important issue.

The security development environments, security configuration management tools, security testing tools are not yet developed for SSBBS. The requirement analysis, design, implementation and evaluation of these tools are important issues.

When implementing and testing the security of SSBBS, it is important to establish a security change control process to have full control of entities and operations. Therefore security change control process development is remaining issue in SSBBS.

It is possible to setup multiple independent organizational networks under a single SSBBS design. In this case, the security setup and management of SSBBS must achieve trust of every party so that the whole system can be administered correctly, efficiently and easily. However, how to construct and maintain SSBBS security for this scenario is an important technical issue.

All sites (geographical location) or application or network do not suit uniform security. Security variation is needed depending on site or application or network. Sometimes, the SSBBS design may have to be modified to suit the security variation. Therefore, how the security variation will be implemented is an important issue in SSBBS.

5. Related Work

The most relevant work to our security analysis of SSBBS was done by Endo et. al. [2] on security in persistently reactive system. In this paper authors defined security requirements and security functions of PCS. They proposed SSB-connector which is embedded with each functional component (not control component) to act upon multiple common roles. However, this work was done before the implementation of P2P based soft system bus based system. And in our paper, we focus on P2P based soft system bus based system. As a result, security issues like threats and environments have got to be different between the papers.

Pfitzmann et. al. have proposed security model for reactive system by means of cryptography [9]. They formulated the requirements in logic and illustrated general model for accepting vulnerabilities. The authors did not consider the persistence, continuity and reliability in their work which are very basic building structure of SSBBS. Besides, apart from cryptography and vulnerability modeling, there are important area like access control, security policy, management, enforcement and considering the security for the system which is distributed among several geographical areas, as a whole.

A well known work on middleware security was done by Demurjian et. al. [10]. In this work, dominant middlewares, namely, CORBA, .NET and J2EE are investigated to compare and contrast from varied perspectives. CORBA and .Net are synchronous middlewares where as the SSBBS is asynchronous. J2EE is asynchronous but there are differences in structure and environment.

6. Conclusion

SSBBS has many brilliant features that make it valuable to implement in large scale distributed PCS. Security analysis for SSBBS was never done in detail that is solved in this paper. Information asset, participants and permissions to information asset in SSBBS and security threats are analyzed. To build secure SSBBS Security requirements are listed up. The important technical issues are illustrated to address the security gap in various aspects. Candidate security tools are mentioned with recommendation. A preliminary design modification is sketched to implement the security.
Complete recommendation of security solutions, design of proposed secure control component, satisfying all security requirements and resolving technical issues are future work.

References

[1] Cheng, J. (2006). Persistent Computing Systems as Continuously Available, Reliable, and Secure Systems. Availability, Reliability and Security, International Conference on, pp. 631-638.
[2] Endo, T., Miura, J., Nanashima, K., Morimoto, S., Goto, Y., and Cheng, J. (2005). Security in Persistently Reactive Systems. 3823, pp. 874-883.
[3] Selim, M. R., Endo, T., Goto, Y., and Cheng, J. (2007). Distributed hash table based design of Soft System Buses. (pp. 1-4). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
[4] Selim, M. R., Goto, Y., and Cheng, J. (2008). Ensuring Reliability and Availability of Soft System Bus. Secure System Integration and Reliability Improvement. pp. 52-59. IEEE.
[5] Selim, M. R., Endo, T., Goto, Y., and Cheng, J. (2006). A Comparative Study Between Soft System Bus and Traditional Middlewares. pp. 1264-1273.
[6] Anderson, R.J. (2008). Security Engineering, Second Edition, Wiley Publishing Inc.
[7] McClure, S., Scambray, J., and Kurtz, G. (2009). Hacking Exposed 6 Network Security Secrets and Solutions, Mc Graw Hill.
[8] Selim, M. R. (2008). A Peer-to-Peer Network Based Middleware for Large-Scale Persistent Computing Systems, Doctoral Theses, Department of Computer Science, Saitama University, Japan.
[9] Pfitzmann, B., Schunter, M., Waidner, M. (2000). Cryptographic Security of Reactive systems, Electronic Notes in Theoretical Computer Science, 32.
[10] Demurjian, S., Bessette, K., Doan, T., and Phillips, C., (2004). "Concepts and Capabilities of Middleware Security," in Middleware for Communications, Q. Mohammed (ed.), John-Wiley, pp. 211-236.

Muhammad Anwarul Azim received the degree of Bachelor of Science (Engineering) from Department of Computer Science and Engineering, Shahjalal University of Science and Technology, Sylhet, Bangladesh in 2000. He got the degree of Master of Engineering from Department of Computer Engineering, Korea Aerospace University, South Korea in 2008. He is a 3rd year PhD student of Department of Information and Computer Sciences at Graduate School of Science and Engineering, Saitama University, Japan. He has been working as a faculty member of Department of Computer Science and Engineering, University of Chittagong, Chittagong, Bangladesh since 2001. His research interests include security of persistent computing system, reliability and information assurance of distributed systems.

Yuichi Goto is an assistant professor of computer science at Graduate School of Science and Engineering, Saitama University in Japan. He received the degree of Bachelor of Engineering in computer science, the degree of Master of Engineering in computer science, and the degree of Doctor of Engineering in computer science from Saitama University in 2001, 2003, and 2005, respectively. His current research interests include relevant reasoning and its applications, automated theorem finding, anticipatory reasoning-reacting systems, and Web services engineering. He is a member of ACM, IEEE-CS, IPSJ, and JSAI.

Jingde Cheng received the Bachelor of Engineering degree in computer science from Tsinghua University in China in 1982, and the Master of Engineering degree and the Doctor of Engineering degree, both in computer science from Kyushu University in Japan, in 1986 and 1989 respectively. He is currently a professor of computer science at Graduate School of Science and Engineering, Saitama University in Japan. Before he joined Saitama University in 1999, he was a research associate (1989-1991), an associate professor (1991-1996), and a professor (1996-1999) at Kyushu University. His current research interests include relevant reasoning, relevant logic and its applications, epistemic programming paradigm for scientific discovery, autonomous evolution of knowledge-based systems, anticipatory reasoning-reacting systems, persistent computing, and information security engineering environment. He is a senior member of ACM, and a member of IEEE-CS, IEEE-SMC, IEEE, and IPSJ.