분산 응용프로그램을 위한 안전한 JavaSpace의 개발

유양우° 이명준
울산과학대학 컴퓨터정보학부
soft@mail.ulsan-c.ac.kr°, mjlee@mail.ulsan.ac.kr

Developing Secure JavaSpace for Distributed Application Programs

Yang-Woo Yu° Myung-Joon Lee
School of Computer Information, Ulsan College

요 약

현재 분산 컴퓨팅 환경에서는 객체를 공유하기 위하여 Jini 서비스인 JavaSpace를 이용하여 다양한 응용 프로그램들이 개발되고 있다. 하지만 JavaSpace의 가장 큰 단점은 누구든지 객체를 저장하고, 또 그 객체를 읽거나 가져갈 수 있도록 설계되어있어 보안성이 매우 취약하다는 것이다.

본 논문에서는 이러한 JavaSpace를 보안등급에 따른 *접근제어*와 SSL 패키지를 이용한 *상호인증* 모듈을 구현하여 JavaSpace의 새로운 보안모델을 제시하였으며, 개발된 안전한 JavaSpace를 이용하여 이동 에이전트간 통신 패러다임으로 적용하였다.

1. 서 론

최근 인터넷의 급속한 성장과 네트워크 기술의 발전으로 다양한 분야에서 분산 애플리케이션들이 계속적으로 등장하고 있다. 다수의 분산된 애플리케이션들은 객체공유, 상호협력(coordination) 및 동적인 통신 서비스를 요구한다. 이러한 요구사항을 만족시키기 위하여 썬 마이크로시스템즈사는 쉽게 사용할 수 있는 JavaSpace 서비스 명세를 제안하였다[1].

현재 분산 컴퓨팅 환경에서는 JavaSpace를 이용하여 다양한 응용 프로그램들이 계속적으로 개발되고 있다. JavaSpace는 Java 기술에 기초로 한 LINDA 모델[2]과 매우 유사하며, 튜플(tuples) 대신에 엔트리(entry) 인터페이스를 구현한 인스턴스를 관리한다. JavaSpace의 프로그래밍 모델은 매우 간결하다. 마치 "shared blackboard"처럼 누구든지 객체를 저장하고, 원하는 객체를 검색하여 그 객체를 읽거나 가져갈 수 있다. 이러한 블랙보드(blackboard) 패러다임은 객체를 공유하는 차원에서는 유용한 서비스로 제공된다. 하지만 분산환경에서 대다수 애플리케이션 프로그램들은 긴밀한 정보를 주고받는다. 특히, 전자상거래 분야에서 객체가 가지는 정보는 고수준의 보안을 요구한다.

본 논문에서는 보안기능이 취약한 JavaSpace를 상호인증 하에 보안등급에 따라 정보를 저장하고 저장된 정보를 검색하여 가져갈 수 있는 새로운 JavaSpace의 보안모델을 제시하였다. 이를 위하여 *접근제어(access control) 모듈*과 *상호인증(authentication) 모듈*을 개발하였다. 개발된 안전한 JavaSpace는 전자상거래 분야에서 널리 응용될 수 있으며, 특히 이동에이전트 시스템[4]에서 에이전트와 에이전트 사이의 상호협력을 위하여 에이전트간 통신을 하고자할 때 안전한 JavaSpace의 유용성을 검증하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 안전한 JavaSpace의 구조에 대하여 자세하게 설명하고, 3장에서는 개발된 안전한 JavaSpace를 이동에이전트 시스템에 적용시킨 사례를 제시하였다. 마지막으로 4장에서는 결론 및 추후 연구 방향에 대하여 살펴본다.

† 본 연구는 정보통신 우수시범학교 지원사업의 지원으로 수행되었음.

2. 안전한 JavaSpace의 구조

2.1 JavaSpace의 소개

Jini 기술은 네트워크 상의 지능형 기기들이나 소프트웨어들이 Jini 네트워크에 접속과 동시에 서비스할 수 있는 기능(Network Plug and Work)을 제공하고, 사용자의 요구에 의한 서비스 요청(Services on Demand)을 처리할 수 있는 새로운 분산 기반 구조를 제공한다[3]. Jini 기반 구조는 서비스 관리자(Lookup Service), Discovery 프로토콜, 그리고 Join 프로토콜로 구성된다. 그리고 분산 응용 프로그램의 개발을 위한 리스(Leases), 이벤트(Events), 그리고 트랜잭션(Transactions) 처리 프로그래밍 모델을 제공하여 분산된 자원의 효율적인 관리와 서비스간의 이벤트 및 트랜잭션 처리를 할 수 있다.

이러한 Jini 서비스 중에 하나인 JavaSpace는 분산 객체의 영속성과 데이터 교환 기능을 지원하는 Jini 서비스로써, 표준 인터페이스를 통하여 객체를 저장(write)하고

가져(take)올 수 있는 기능과 저장된 객체를 템플릿을 사용하여 탐색할 수 있는 기능 등을 제공한다. 또한 JavaSpace는 하나의 객체를 다중 사용자가 동시에 접근하려고 할 때 고려해야 할 동시성제어(concurrency control)를 지원한다. 다음의 표1은 JavaSpace에서 제공하는 인터페이스를 소개하고 있다.

표 1. JavaSpace 인터페이스

메소드	설 명
write	주어진 엔트리(entry)를 JavaSpace의 저장소에 저장한다.
read	주어진 템플릿(template)과 매칭되는 엔트리를 읽는다.
readIfExists	주어진 템플릿과 매칭되는 엔트리를 읽는다. 매칭되는 엔트리가 없다면 블록(block)되지 않고 null을 반환한다.
take	주어진 템플릿과 매칭되는 엔트리를 읽고, JavaSpace 저장소에서 제거 한다.
takeIfExists	주어진 템플릿과 매칭되는 엔트리를 읽고, JavaSpace 저장소에서 제거 한다. 매칭되는 엔트리가 없다면 블록(block)되지 않고 null을 반환한다.
notify	주어진 템플릿과 매칭되는 엔트리가 JavaSpace에 저장되면 지정된 객체 에게 그 사실을 알려준다.
snapshot	원래 엔트리의 snapshot을 포함하는 새로운 엔트리 객체를 반환한다.

2.2 안전한 JavaSpace의 설계

본 논문에서 제안하는 안전한 JavaSpace는 기본적으로 썬 마이크로시스템즈사에서 제공하는 API를 기반으로 설계되었으며, 보안기능이 요구되는 접근에 대해서는 상호인증 하에 보안등급에 따라 정보를 저장하고, 저장된 정보를 읽고 가져갈 수 있도록 안전한 통신정책을 제공하는 시스템이다.

안전한 JavaSpace의 기본적인 시스템 구조는 *접근제어(access control)*와 *상호인증(authentication) 모듈*이다. 먼저, 접근제어는 보안등급을 설정하는 *SecurityLevel 인터페이스*를 구현하고, 보안등급에 따라 접근을 허용하는 JavaSpace 서버 측의 데몬 형태로 수행되는 *SecureJavaSpace 클래스*를 정의한다. 인증된 멤버만이 스페이스에 접근할 수 있다.

이와 같이 개발된 안전한 JavaSpace를 이용하여, 이동 에이전트간 보안기능이 추가된 통신을 지원하는 이동 에이전트 시스템에 적용하여 그 유용성을 검증한다.

2.3 안전한 JavaSpace의 기본모듈

안전한 JavaSpace 시스템의 기본 모듈은 *SecurityLevel 인터페이스*와 데몬 형태로 동작하는 *SecureJavaSpace 클래스* 두 가지로 구성된다. *SecurityLevel 인터페이스*에서 정의된 보안등급은 *Admin, User, Anony* 세 가지로 구분된다. 보안등급이 Admin인 경우에는 JavaSpace내의 모든 엔트리를 접근(read, write, take)할 수 있다. 그리고, User인 경우에는 제한적인 접근(read, write)이 가능하다. 마지막으로 보안등급이 Anony인 경우에는 "read" 접근만이 가능하도록 설계 구현한다.

JavaSpace에서 데몬 형태로 동작하는 *SecureJavaSpace 클래스*는 클라이언트에서 요청하는 메소드에 따라 다른 서비스를 제공한다. 일반적인 JavaSpace의 기본 인터페이스를 요청하면 보안기능이 없는 JavaSpace의 엔트리를 제공하고, 기본 인터페이스의 메소드에 보안등급(Admin, User, Anony)이 인자로 추가된 메소드를 요청하면 보안기능이 제공되는 안전한 JavaSpace의 엔트리를 제공한다.

클라이언트 인증과 서버 인증은 RMI와 TCP 계층사이에 SSL(Secure Socket Layer) 패키지를 이용하여 해결할 수 있다[4]. SSL은 대칭 또는 비대칭적인 암호화 방법을 실제 사용할 수 있도록 작성한 산업 표준 프로토콜로서 기밀성과 데이터 무결성 그리고 클라이언트/서버의 상호 인증을 제공한다.

```
// Communicator.java
   SSLServerSocket ss = new SSLServerSocket(port);
   while(true) {
     SSLSocket s = (SSLSocket)ss.accept();
     ProxyObject po = new ProxyObject(s);
     po.start();
   }
// ProxyObject.java
   public void run() {
     try {
                s.startHandshake();
                s.waitForHandshake();
                InputStream is = s.getInputStream();
                OutputStream os = s.getOutputStream();
     } catch(Excetpion e) {
                e.printStackTrace();
     }
     Agent_Message agent_Message =
                 (Agent_Message)is.read();
   }

// ProxyClient.java
   public void run() {
     SSLSocket s = new SSLSocket(host, port);
     s.startHandshake();
     s.waitForHandshake();
     OutputStream os = s.getOutputStream();
     InputStream is = s.getInputStream();
     os.write(agent_Message);
     os.flush();
   }
```

그림1. SSL API를 이용한 프로그래밍

본 시스템에서는 *Protekt 3.0 SSL* 패키지에서 제공하는 API를 이용하여 JavaSpace의 보안모델에서 요구하는 보안요

소를 만족시켰다[4]. 다음은 SSL을 이용한 JavaSpace의 Communicator와 Proxy 객체에 대한 예를 보여 주고 있다.

3. 안전한 JavaSpace의 적용

보안기능을 제공하는 JavaSpace를 구현함으로서, 그 적용분야는 매우 다양하다. 특히, 분산 애플리케이션 영역에서 객체를 저장하고 검색할 때, 보안기능을 갖춘 JavaSpace는 정보의 안전성을 크게 증대시킬 수 있다.

이전의 연구[8]에서 *이동에이전트 시스템*의 개발에 대하여 연구를 계속 진행해왔었다[7]. *이동에이전트*의 개념은 사용자가 정의한 이동경로를 따라 여러 이동에이전트 시스템들로 자발적으로 이동하여 자신의 작업을 수행하고, 그 결과를 사용자에게 되돌려주는 패러다임이다[5]. 하지만, 대다수 이동에이전트 시스템들은 에이전트와 에이전트간 직접 통신기능을 지원하지 않고 있다. 이를 지원하는 소수의 시스템들은 일반적으로 에이전트간 통신 시, 이동에이전트 시스템을 통하여 간접적 보안기능이 취약한 상태에서 수행된다. 이러한 문제점을 해결하기 위하여, 본 논문에서는 안전한 JavaSpace를 이용하여 에이전트간 상호협력이 가능한 통신 패러다임을 제시하고, 실제 개발된 이동에이전트 시스템에서 안전한 JavaSpace의 유용성을 검증할 것이다.

이동에이전트간 통신 정책은 그림2 에서 설명하는 것처럼 메시지 전달 방식에 의해서 수행된다. 송신자 에이전트는 메시지 객체를 생성하고, 유일한 이름을 가진 수신자 에이전트의 이름으로 메시지를 작성한다. 작성된 메시지 객체는 JavaSpace에 직접 접근하기 이전에 데몬 프로세스와 접근하여 보안등급을 검사한다. 그리고, 보안등급에 따라 메시지가 저장된다.

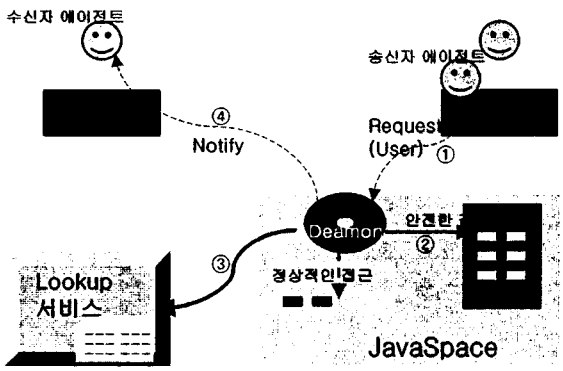

그림 2. 안전한 JavaSpace를 이용한 이동에이전트 간 통신 정책

메시지가 정상적으로 저장되면, 데몬 프로세스는 수신자 에이전트의 이름으로 Jini[2][5]의 LookUp 서비스를 이용하여 수신자 이동에이전트의 위치정보를 얻을 수 있다. 그리고 그 정보를 이용하여 수신자 에이전트에게 자신의 메시지가 도착했다는 이벤트를 통지한다.

4. 결론 및 추후연구

JavaSpace의 프로그래밍 모델은 누구든지 객체를 저장하고, 원하는 객체를 검색하여 그 객체를 읽거나 가져갈 수 있도록 설계되어있다. 이러한 블랙보드(blackboard) 패러다임은 객체를 공유하는 차원에서는 유용한 서비스로 사용되지만, 분산환경에서 대다수 애플리케이션 프로그램들은 긴밀한 정보를 주고받기를 원한다.

본 논문에서는 보안기능이 취약한 JavaSpace를 상호인증 하에 보안등급에 따라 정보를 저장하고 저장된 정보를 검색하여 가져갈 수 있는 안전한 JavaSpace를 설계 구현하였다. 기본적인 시스템 구조는 보안등급에 따른 *접근제어(access control)* 모듈과 SSL 패키지를 이용한 *상호인증(authentication) 모듈*이다.

개발된 안전한 JavaSpace를 이용하여 이동에이전트 시스템의 기능을 확장하고, 이동에이전트를 이용한 전자상거래 응용프로그램을 개발하여 안전한 JavaSpace의 유용성을 검증하였다.

추후 연구과제는 현재 SSL 패키지를 이용한 방식을 자바에서 제공하는 JDK1.4 내에 라이브러리 형태로 존재하는 JAAS(Java Authentication and Authorizing Service)와 JERI(Jini Extensible Remote Invocation)를 이용한 방식으로 제공할 예정이다.

[참고문헌]

[1] Sun Microsystems, Inc. JavaSpacesTM Service Specification, Version 1.1, October 2000.

[2] Sun Microsystems Inc., "Jini Technology Core Platform Specification", 2000.10.

[3] Jan Newmarch, "Jan Newmarch's Guide to JINI Technologies," http : //jan.netcomp.monash.edu.au /java/jini/tutorial/Jini.xml.

[4] Forge Information Technology, "Protekt Encryption 3.0 Programming Guide," 1999

[5] Scott Oaks and Henry Wong, "Jini in a Nutshell", Oreilly Press, March 2000.

[6] Alfonso Fuggetta, Gian Pietro Picco, Giovanni Vigna, "Understanding Code Mobility," IEEE Transaction On S/W Engineering, Vol.24, No.5, May, 1998.

[7] 구형서, 윤형석, 김진홍, 유양우, 문남두, 이명준, "Jini 기반의 이동 에이전트 시스템" 한국정보과학회 가을 학술발표논문집 Vol.28, No.1, 2001.

[8] 김진홍, 구형서, 윤형석, 안건태, 유양우, 이명준, "JMoblet: Jini 기반의 이동에이전트 시스템.", 한국정보처리학회논문지 B 제8-B권 제6호, 2001.12.