이동 에이전트 시스템 구현 스택의 제안

신정화*, 신 원*, 이경현**

*부경대학교 대학원 전자계산학과, **부경대학교 전자컴퓨터정보통신공학부

A Proposal of Mobile Agent System Implementation Stack

Jung-Hwa Shin*, Weon Shin*, Kyung-Hyune Rhee

* Department of Computer Science Pukyong Univ.
**Division of Electronic, Computer and Telecommunication Engineering Pukyong Univ.

요 약

본 논문에서는 이동 에이전트 시스템의 특징을 살펴보고, 현재 다양한 네트워크 기술에 응용되고 있는 Java 기술을 적용한 이동 에이전트 시스템을 비교·분석하였다. 이를 기반으로 안전한 이동 에이전트 시스템 구현을 위한 이동 에이전트 시스템 구현 스택(MASIS)을 제안하고 각 계층별 시큐리티 요구사항에 대하여 논의한다.

I. 서론

이동 에이전트는 주어진 작업을 수행하기 위해 네트워크로 연결된 여러 시스템 사이를 이동하는 지능적인 프로그램을 의미한다. 이러한 이동성은 이동 에이전트 작성을 위해 사용하는 이동 코드에 그 기반을 두고 있으므로, 이동 에이전트의 보안 문제는 궁극적으로는 여러 시스템에서 실행 가능한 이동 코드 자체에 대한 보안 문제에 기반한다.

실제 이동 코드 제작은 프로그램을 컴퓨터 시스템 사이에서 교환하는 것을 의미한다. 또한 실행 환경의 다양성을 위해 가상 머신 상에서 해석되도록 구성된다. 즉, 이동 코드 기술에 기반한 이동 에이전트를 이용하여 이기종 시스템의 분산 컴퓨팅 환경을 구성하는 것이 가능하다. 그러나 이동 에이전트는 코드 자체가 여러 호스트 사이를 이동하도록 구현되므로 이동 코드 자체와 이동 코드를 받아들여 수행하는 호스트측 모두에게 새로운 보안 문제를 야기한다.

따라서, 본 논문에서는 호스트측과 이동 에이전트 보호를 위한 시스템 구현 스택을 제안한다. 2장에서 Java 기반 이동 에이전트 시스템의 특징과 시큐리티에 대해서 살펴보고 3장에서는 안전한 이동 에이전트 시스템을 위한 구현 스택을 제안한 후 4장에서 결론을 유도한다.

II. Java 기반 이동 에이전트 시스템

현재 많은 이동 에이전트 시스템들이 개발되고 있으며 다양한 활용 분야를 위해 이동 에이전트 시스템에 대한 연구가 활발히 진행되고 있다. 실제 이를 위해 Telescript[1], Agent Tcl[2] 등의 이동 에이전트 전용 언어가 개발되었으나, 최근에는 플랫폼 독립성을 장점으로 다양한 환경에 적용되고 있는 Java를 기반으로 하는 시스템들이 다수를 차지하고 있다. 따라서, 본 장에서는 분산 처리 및 네트워크 개발 언어로 자리 잡고 있는 Java 언어로 개발된 이동 에이전트 시스템을 중심으로 각 시스템의 특징과 시큐리티를 비교한다.

1. 각 이동 에이전트 시스템의 특징

1) Aglets

IBM에서 개발된 Aglets[3]은 위치 기반 URL과 Java 객체 직렬화를 사용하여 이동성을 구현한다. 또한 Aglet은 메시지 전달만을 통해서 상호 통신이 가능하며 제한된 시큐리티만을 지원한다.

2) Voyager

Voyager[4]는 ObjectSpace에 의해 개발되어, 리모트에서 위치 기반 액세스를 제공하는 가상 참조를 생성한다. 에이전트 통신은 가상 참조 상에서

메소드 호출을 통하여 가능하며, 에이전트가 그룹에 계층적으로 모여 있기 때문에 멀티캐스트도 가능하다.

3) Concordia

Mitsubishi에서 개발된 Concordia[5]는 Java 객체 직렬화를 사용하여 이동성을 제공한다. Concordia는 비동기 이벤트 신호처리 제공 및 특정 그룹 협동작업 메커니즘을 제공한다. 또한 암호 프로토콜을 사용하여 에이전트 상태를 계획하고, 사용자에 기반한 ACL을 정적으로 사용하여 자원 사용을 계획할 수 있다.

4) Ajanta

University of Minnesota에서 개발된 Ajanta[6]는 Java 객체 직렬화를 사용하여 에이전트 이동성을 제공하고 다른 에이전트의 간섭을 막기 위해서 격리된 보호 영역에서 각 에이전트들이 수행되도록 한다. 특히, Ajanta는 에이전트 상태를 보호하기 위하여 에이전트의 소유자가 에이전트의 상태에 대한 수정을 검출하도록 하는 암호 메커니즘을 제공한다.

5) Mole

University of Stuttgart에서 개발된 Mole[7]은 Sandbox 시큐리티 모델을 도입하고, JNI로 서비스 에이전트를 작성하여 에이전트 시스템 내부의 자원, 제어를 담당하도록 하였다. 로컬 디렉토리 서비스 및 자원 관리 서비스를 제공한다.

6) MOA

MOA[8]는 Open Group Research Institute에서 개발되어, 에이전트 이전은 소켓을 이용하여 에이전트의 상태를 목적지에 전송하고 새로운 인스턴스를 생성함으로써 이루어진다. 이름 기반의 상호통신 및 이전이 가능한 Agent와 다른 에이전트 및 장소와 통신 가능한 Place, 각 에이전트 제어를 위해서 Monitor, Logger, Mover 등을 제공한다.

2. 시큐리티 분석 및 비교

인터넷과 같은 공개 네트워크는 안전하지 않으므로 현재 이동 에이전트 시스템 환경에서는 다음의 예와 같은 여러 가지 시큐리티 문제가 발생할 수 있다.

- 사용자의 도청 및 서버에서 에이전트 실행에 의한 에이전트 내에 포함된 중요 정보의 노출
- 서버에 의해 에이전트의 코드, 제어 흐름, 실행 결과 등의 변경
- 에이전트에 의해 서버의 중요 정보 노출
- 에이전트가 서버 자원을 독점하여 서버에 DoS 공격

표 1: 에이전트 시스템 시큐리티 비교

에이전트 시스템	안전한 통신	서버 자원 보호	에이전트보호
Aglets	×	trusted & untrusted 정적인 액세스 제어	×
Voyager	SSL 사용	native & foreign SecurityManager를 확장	×
Concordia	SSL 사용	SecurityManager 사용 에이전트 소유자 기반 동적으로 구성된 ACL	자원 액세스 메커니즘사용
Ajanta	ElGamal과 DSA 사용	에이전트 소유자 기반	상태와 코드 위조 감지 메커니즘
Mole	×	Sandbox 시큐리티 모델	×
MOA	×	Sandbox 시큐리티 모델	×

따라서, 다양한 공격에 대하여 안전한 이동 에이전트 시스템 동작을 보장하기 위한 메커니즘이 필수적이다. 즉, 데이터와 코드를 보호하기 위한 비밀성 메커니즘, 참여자의 신분 확인을 위한 인증 메커니즘, 서버 자원에 대한 액세스 제어를 위한 허가 메커니즘 등이 이에 포함된다. 표 1은 여러 이동 에이전트 시스템들이 제공하는 시큐리티 측면을 제안된 시스템 별로 비교한 것이다.

III. 안전한 이동 에이전트 시스템을 위한 구현 스택의 제안

살펴본 바와 같이 많은 에이전트 시스템이 개발되어 왔으나 인터넷과 같이 공개된 네트워크 상에서 안전한 이동 에이전트 시스템을 동작하기에는 다소 무리가 있다. 따라서, 이동성을 가장 큰 장점으로 하는 이동 에이전트 원래의 목적을 제대로 수행하기 위해서는 이동 에이전트 시스템에 대한 시큐리티 측면이 반드시 고려되어야 한다. 이를 통하여 안전한 에이전트 시스템을 구성하고 이동 에이전트의 투명성을 확보함으로써 효율적이면서도 편리한 컴퓨팅 환경을 사용할 수 있게 될 것이다. 본 장에서는 안전한 이동 에이전트 시스템의 설계 및 구현 등에 대하여 다룬다.

1. 이동 에이전트 시스템 구현 스택

이동 에이전트 시스템은 명령 수행을 위한 데이

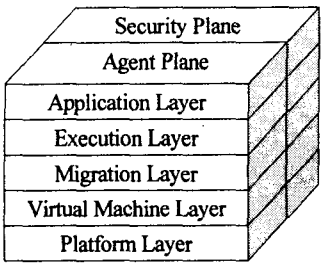

그림 1: 이동 에이전트 시스템 구현 스택(MASIS)

터 정보, 실행 정보, 상태 정보 등이 수록된 이동 코드에 기반하므로, 이동 코드가 실행하기 위한 시스템 환경 구축이 필수적이다. 이를 위해서는 프로그래밍 및 실행을 위한 플랫폼이 구성되어야 하고, 이동 에이전트의 이전을 위한 네트워크 환경도 구축되어야 한다. 또한, 이동 에이전트 시스템을 보호하기 위해서는 암호 기술 등이 구현된 정보보호 기술이 필수적으로 포함되어야 한다. 본 장에서는 안전하고 효율적인 이동 에이전트 시스템 구축을 위한 "이동 에이전트 시스템 구현 스택(MASIS, Mobile Agent System Implementation Stack)"을 제안하고 각 계층별 세부 구성 요소의 기능을 설명한다. 그림 1은 본 논문에서 제안하는 이동 에이전트 시스템 구현 스택을 보여준다. 구현 스택은 에이전트 구현을 위한 평면(Plane)과 시큐리티 제공을 위한 평면(Plane)으로 구성되며 각 평면은 5개의 계층(Layer)으로 나뉘어진다.

이동 에이전트 시스템은 자체의 이동성으로 인하여 에이전트를 실행하는 호스트, 이동 코드를 실행하는 에이전트, 에이전트끼리의 통신 등에 대해서도 시큐리티를 유지하여야 하므로 단순한 전송 계층에서의 암호화 및 전자서명 같은 서비스만으로 전체 시스템을 보호하기에는 무리가 있다. 따라서, 에이전트 시스템의 구현 환경에 맞추어 각 시큐리티 단계를 체계적으로 분류하고 이에 따른 별도의 시큐리티 서비스가 필수적이다. 이를 위하여 MASIS에서는 시큐리티 서비스를 위한 평면인 "Security Plane"을 두고, 해당하는 각 계층에서의 시큐리티를 별도로 정의하고 있다. 표 2는 각 계층별로 구현되어야 할 시큐리티 서비스를 보여준다.

2. 각 계층별 시큐리티 서비스

1) 플랫폼 계층(Platform Layer)

"플랫폼 계층"에서 시큐리티는 물리적인 시큐리티에서부터 안전한 운영체제 및 네트워크, 시스템 자원 보호와 같은 플랫폼 시큐리티를 모두 포함한다. 기본적으로 시스템 보호를 위한 Secure OS, 네트워크 상의 데이터 보호를 위한 VPN, SSL/TLS 등이 여기에 포함된다.

표 2: 각 계층별 시큐리티 서비스

계층	시큐리티 서비스	적용 기술 예
Application Layer	Identification Privacy	암호화 전자서명
Execution Layer	Privacy Integrity	암호학적 추적 수행결과 보호
Migration Layer	Authentication	안전한 이전
Virtual Machine Layer	Authorization Access Control	Sandbox PCC
Platform Layer	Access Control Confidentiality	Secure OS VPN SSL/TLS Signed Code

PCC : Proof Carrying Code
VPN : Virtual Private Network
SSL : Secure Socket Layer
TLS : Transport Layer Security

2) 가상 기계 계층(Virtual Machine Layer)

"가상 기계 계층"에서 시큐리티는 임의의 플랫폼 가상 머신 상에서 동작하는 에이전트를 제어하고 모니터링함으로써 악의적인 에이전트로부터 호스트를 보호하는 것이다. Java에서는 Sandbox 모델[9]을 적용하기 위해 Class Loader, Bytecode Verifier, Security Manager를 확장하거나 수정해서 이를 수행한다. 부가적으로 시스템에서의 안전한 동작에 대한 증명을 부여하는 PCC[10]를 사용할 수도 있다.

3) 이전 계층(Migration Layer)

"이전 계층"에서 시큐리티는 안전한 플랫폼을 기반으로 에이전트 이전을 안전하게 수행하기 위한 것으로, 에이전트의 근원지 호스트와 목적지 호스트 간의 상호 인증이 필수적이다. 또한 에이전트의 실행 코드, 데이터, 실행 상태가 이전된 이후에도 계속해서 원래의 작업을 수행해야 하므로 이를 위한 방안이 마련되어야 한다. Java 기반 에이전트 시스템에서는 RMI와 객체 직렬화를 도입하여 에이전트 코드, 데이터, 실행 상태에 대한 암호화 및 전자 서명을 함께 사용하여 이전을 수행할 수 있다.

4) 실행 계층(Execution Layer)

"실행 계층"에서 시큐리티는 악의적인 호스트로부터 에이전트를 보호하기 위한 것으로 에이전트

실행을 방해받거나 위조되지 않도록 하고 데이터의 불법적인 수정을 막는 것이다. 따라서, 안전한 에이전트의 동작을 위한 비밀성, 인증성 등이 필수적으로 구현되어야 한다. 실제 시스템에서는 호스트로부터 실행을 보호하기 위하여 Code Obfuscation, 수행 결과 보호 방안, Cryptographic Trace 등이 적용가능하다[11].

5) 응용 계층(Application Layer)

"응용 계층"에서 시큐리티는 정당한 사용자만이 에이전트 시스템에 접근하도록 허용하여 에이전트에 의해 수행된 결과를 불법적으로 접근할 수 없도록 하는 것이다. 신분확인을 위한 전자 서명 및 데이터 비밀성을 위한 암호화 방안, 정당한 에이전트 사용자에 대한 프라이버시 보호도 함께 제공되어야 한다.

안전한 이동 에이전트 시스템을 구현하기 위해서는 시스템 동작에 대한 체계적인 분석이 필요하고 그에 따른 필수적인 시큐리티 요구 사항을 정립한 후 적절한 정보보호 기술 구현이 함께 이루어져야 한다. 또한 응용분야에 따라 이동 에이전트 시스템에서의 특정 계층에 초점을 맞추어 그에 해당하는 시큐리티를 강화할 수도 있다.

IV. 결론

본 논문에서는 최근 네트워크 개발 언어로 각광받고 있는 Java를 기반으로 개발된 이동 에이전트 시스템의 특징을 살펴보고 시큐리티 측면을 비교·분석하였다. 이를 기반으로 이동 에이전트 시스템의 설계 및 구현에 있어 안전성을 실현하기 위한 구현 스택 MASIS(Mobile Agent System Implementation Stack)를 제안하고 각 계층별 시큐리티 요구 사항을 정립하였다.

현재 네트워크 상에서 동작하는 에이전트들은 기능별로 세분화되어 업무에 활용 가능한 전자비서 에이전트, 전자상거래에서의 대리자인 상거래 에이전트, 정보검색을 위한 로봇 에이전트, 분산처리를 위한 협력 에이전트 등이 등장하고 있다. 그중 이동 에이전트 시스템에 있어 이동성은 시스템에 대한 통합을 쉽게 해주고 여러 시스템에 있어 융통성을 제공하는 반면 새로운 보안 문제를 발생시키고 있으며, 이로 인하여 이동 에이전트 시스템 구현에 있어 큰 장애요소로 등장하고 있다. 따라서, 이동 에이전트가 여러 다양한 분야에 직접적으로 활용되기 위해서는 이동 에이전트 시스템에 새롭게 등장하는 시큐리티 문제를 다루고, 안전한 에이전트 기반 시스템을 구현하기 위한 깊이 있는 연구가 필요하다. 특히, 전자상거래의 보급에 있어서 이들 에이전트 시스템은 매우 중요한 위치를 차지하고 있으므로 나날이 고도화되고 지능화되는 다양한 사이버 공격에 대해서도 안전한 이동 에이전트 시스템 구축에 대한 연구가 활발히 이루어져야 할 것이다.

참고문헌

[1] J.E.White, "Telescript Technology: The Foundation for the Electronic Marketplace," General Magic, Inc., Sunnyvale, CA, 1994

[2] R.S.Gray, "Agent Tcl: A Transportable Agent System," Proceedings of the CIKM Workshop on Intelligent Information Agents, Baltimore, MA, 1995

[3] http://www.aglets.org/

[4] http://www.objectspace.com/products/voyager/

[5] http://www.concordiaagents.com/

[6] http://www.cs.umn.edu/Ajanta/

[7] http://mole.informatik.uni-stuttgart.de/

[8] http://www.opengroup.org/

[9] http://java.sun.com/

[10] G.C.Necula, and P.Lee, "Safe, Untrusted Agents Using Proof-Carrying Code," In: G.Vigna (Ed.), Mobile Agents and Security, Springer-Verlag, Lecture Notes in Computer Science 1419, pp. 61-91, 1998

[11] 신원, 이경현, "이동 에이전트 실행 보호를 위한 암호학적 추적 방안", 「한국통신정보보호학회 논문지」, Vol.11 No.3, pp.71-78, 2001.

[12] S.Oaks, "Java Security," O'Reilly & Associates, Sebastopol, CA, 1998

[13] B.Schneier, "Applied Cryptography", 2nd, John Wiley & Sons, 1996

[14] A.Menezes, P.van Oorschot, and S.Vanstone, "Handbook of Applied Cryptography", CRC Press, Boca Raton, FL, 1996

[15] A.H.W.Chan, M.R.Lyu, "The Mobile Code Paradigm and Its Security Issues," World Wide Web: Technologies and Applications for the New Millenium, CSREA Press, pp.353-357, 2000

[16] N.Karnik, "Security in Mobile Agent Systems," Ph.D. Thesis, Department of Computer Science and Engineering, University of Minnesota, 1998