□신기술해설□

이동코드와 보안문제

박 종 열† 신 욱†† 이 동 익†††

◈ 목 차 ◈

1. 서 론

최근 인터넷 사용자의 폭발적 증가에 따라 인터넷 관련 사업의 영역 역시 급격히 증가하고 있다. 이동에이전트는 인터넷 환경에서 다양하게 정의되는 응용들을 수용하기 위해 제안, 연구되고 있는 패러다임 중의 하나로 최근 많은 주목을 받고 있는 기술이다.

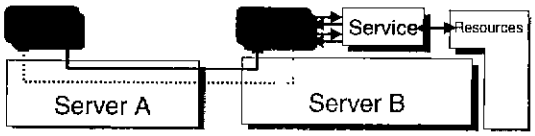
이동에이전트 시스템이란 '네트워크로 연결된 독립적인 다수의 소프트웨어 프로세서(에이전트)가 대등한 관계를 유지하며 하나의 유기적인 집합체로서의 기능을 수행하는 시스템'으로, 현재의 클라이언트 서버 시스템의 문제점을 해결할 수 있는 새로운 분산처리 패러다임으로 인식되고 있다. 이동에이전트 시스템의 장점은 다양하고 이질적인 시스템을 손쉽게 통합할 수 있다는 것과, 상황 변화에 따라 프로그램이 동적으로 대처할 수 있도록 유연성을 제공한다는 것이다. 특히 에이전트 생성자 역할을 하는 에이전트 서버가 전체 시스템의 동작을 임의로 정의하고 변경할 수 있다는 특징은 이동에이전트 시스템만이 갖는 장점이다[1][2][3].

이동에이전트의 이동성은 유연성과 시스템 통합 용이성을 제공해 주는 반면 새로운 보안문제를 야기하였으며, 이는 이동에이전트의 실용화에 큰 장애요소로 작용하고 있다. 이동에이전트의 이동성은 ActiveX, Java등으로 대표되는 이동코드 기술에 그 바탕을 두고 있으며 따라서 이동에이전트의 보안 문제는 이동코드의 보안 문제로 귀결된다고 볼 수 있다. 본 고에서는 이동에이전트 시스템을 중심으로 이동코드 기술에 의해 새로이 발생하는 보안문제와 현재까지 연구되고 제안된 대응책에 관해서 해설한다. 그러나 지면관계상 이에 관한 모든 연구 결과를 망라하는 것은 불가능한 일일 것이다. 따라서 여기에서는 대표적인 대응 기술에 관하여 개략적인 소개만을 할 예정이며, 기술적인 내용을 상세히 알고자 하는 독자는 참고문헌을 참조할 것을 권한다. 앞으로, 2장에서는 이동에이전트 시스템과 이동에이전트 시스템에서 발생하는 보안문제에 관해서 설명하고, 3장과 4장에서는 이동에이전트 보안에 관한 최근의 연구동향을 설명한다. 그리고 5장에서 결론으로 해설을 맺는다.

2. 이동에이전트 시스템과 보안문제

2.1 이동에이전트

† 준 회 원 : 광주과학기술원 정보통신공학과 박사과정
†† 준 회 원 : 광주과학기술원 정보통신공학과 박사과정
††† 종신회원 : 광주과학기술원 정보통신공학과 부교수



(그림 1) 이동에이전트 개략

이동에이전트는 주어진 작업의 수행을 위하여, 네트웍으로 연결된 호스트 사이를 스스로의 판단하에 이동하는 지능적인 프로그램을 의미한다. 이동에이전트는 이동성(mobility) 이외에도 지능성(intelligence), 사회성(sociality), 자율성(autonomy), 결단성(decidability) 등으로 일컬어지는 다양한 특성을 지닌다. (그림 1)은 이동에이전트가 동작하는 모습을 간략히 보여주는 그림으로 에이전트가 원격 호스트로 이동하여 코드를 수행하는 모습을 나타내고 있다.

2.2 이동에이전트의 보안문제

이미 서론에서 언급한 바 있듯, 코드의 이동은 이동하는 코드와 이동코드를 받아들여 수행하는 양측 모두에게 새로운 보안문제를 야기하는데, 이동에이전트 시스템에서 이들 보안 문제는 다음과 같이 크게 셋으로 요약될 수 있다.

- 에이전트 클론 생성[1]: 전자 상거래와 같이 이동에이전트가 중요한 정보를 운반하는 응용 어플리케이션에서 에이전트 클론의 생성은 보안상의 심각한 위협요소로 작용한다. 원래의 이동경로에서 벗어난 에이전트 클론은 중요한 정보를 유출시킬 수 있으며, 프로세스를 한 번 이상 수행하여 데이터 무결성에 위해를 가할 수도 있다.
- 악의를 가진 호스트로부터의 이동에이전트 보호: 원칙적으로 이동에이전트의 수행은 제삼자에 의해 간섭을 받아서는 안되며, 이동에이전트 내에 저장된 중요한 정보와 수행코드 등이 이동에이전트 외부에 노출되어서도 안된다. 그러나 이동에이전트 서버 즉, 호스트는 이동에이전트 코드의 수행을 책임지는 주체로서, 에이전트에게 수행환경을 제공하고, 에이전트 내부의 데이터를 참조, 갱신하기 위하여, 궁극적으로는 이동에이전트 내부의 모든 공간에 접근할 수 있다. 따라서 호스트가 악의를 갖고 이동에이전트에 위해를 가하고자 하는 경우, 에이전트를 보호하는 것은 매우 어렵다.
- 악의를 가진 이동에이전트로부터의 호스트 보호: 호스트의 입장에서 볼 때, 이동에이전트는 외부에서 유입되는, 신뢰할 수 없는 프로그램이다. 호스트 내에서 수행되는 이동에이전트는 바이러스 프로그램처럼 호스트의 중요한 정보를 임의로 삭제 또는 변경할 수 있으며, 자원을 독점하여 다른 이동에이전트의 수행이 불가능하도록 피해를 입힐 수도 있기 때문이다. 따라서 호스트는 에이전트 수행 동안 자원 접근을 감시하고 통제함으로써 안전한 이동에이전트 서버의 수행을 보장할 수 있도록 대응 방안을 마련해야 한다.

3. 이동에이전트를 위한 보안 모델

이 절에서는 2.2절에서 언급한 이동에이전트의 보안 문제 중, 악의를 가진 호스트로부터의 에이전트 보호문제와 관련하여 이제까지 제안, 연구되어온 이동코드 보안 모델 및 보호 방법들을 소개하고자 한다. 이동에이전트의 안전한 수행을 보장하고자 하는 노력은 안전한 에이전트 코드 수행을 위한 노력과 에이전트 수행 데이터 보호를 위한 노력, 두 범주로 나누어 생각해 볼 수 있다.

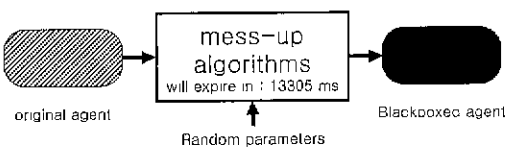
[1] 하나의 에이전트 프로그램이 두 개 이상으로 복사되어 수행되는 경우 복사된 에이전트를 클론 에이전트라 한다.

3.1 안전한 에이전트 코드 수행

이 방법은 에이전트 서버가 에이전트 코드를 수행하는 동안 자신이 수행하는 코드의 의미나 값을 알 수 없도록 하여, 안전한 에이전트의 수행을 보장하고자 하는 방법이다.

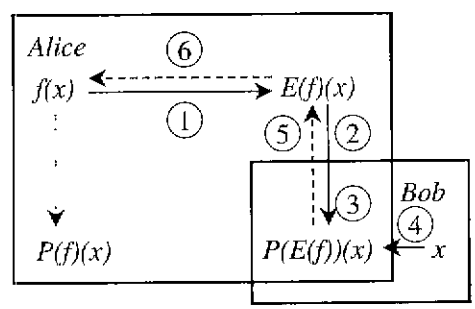
이동에이전트 코드가 수행되는 시점에서 코드에 비밀성을 부여하기 위하여 지금까지 제안된 대표적인 방법으로는 Blackbox Security[4]와 Mobile Cryptography[5]가 있다.

- Blackbox Security: 이동에이전트 공격은 서버의 에이전트 코드 해석에서부터 시작한다는 가정 하에, 이동에이전트의 코드를 마구 뒤섞어서(mess-up) 서버가 해석하기 힘든 코드의 형태로 치환하는 방법이다. 예를 들어 price 라는 변수가 가격을 저장한다고 하면, Blackbox Security는 price를 A와 B라는 평범한 이름의 변수 두 개로 나누고, A는 price의 홀수 비트를 B는 짝수 비트를 나타내도록 바꾸어 서버가 연산을 수행하는 동안, 변수의 의미를 파악하지 못하도록 하는 방법이다. 물론 연산과정도 뒤섞는다. Blackbox 모델((그림 2) 참조)은 이처럼 code mess-up 알고리듬을 사용하여 에이전트 코드를 변환하는 외에 시간 제약(time-out)을 부여하여 코드를 뒤섞는 법칙이 분석, 파악되는 것을 막고 있다[4]. 현재 Blackbox 모델과 관련하여 진행되고 있는 연구로는 code mess-up 알고리듬 분석 시도에 대응하기 위한 protocol의 개발이 있다[6].
- Mobile Cryptography: 이 방법은 이동에이전트의 함수를 암호화하여 수행하는 방법이다. 이 때의 암호화란, 원래의 함수와 수학적으로는 같은 의미를 가지나 형태는 다른 제 2의 함수를 찾아 원래 함수 대신 치환함으로써, 코드 수행자에게 원래 함수의 형태를 보여주지 않고 원하는 연산을 수행하는 과정을 의미한다[5]. Mobile Cryptography 모델이 실용화되기 어려운 이유는, 이동에이전트 코드의 암호화를 위해 어떤 함수와 형태는 다르나 동일한 수행 결과를 반환하는 준동형(Homomorphic) 함수((그림 3) 참조)들을 찾기가 어렵기 때문이다. 따라서, Mobile Cryptography 모델은 다양한 이동에이전트의 동작을 암호화 하기에는 한계가 있다.



(그림 2) 시간 제약이 있는 Blackbox 방법

위에서 열거한 에이전트 코드에 비밀성을 부여하고자 하는 방법은, 데이터 암호화를 사용하던 기존의 방법에 비하여 획기적인 접근 방법이지만, 코드 수행이 종료된 후 에이전트가 저장하고 있는 데이터의 무결성을 파괴하고자 하는 시도가 있을 수 있음을 가정할 때, 에이전트의 안전한 수행을 보장하는 완전한 해결책은 될 수 없다. 따라서, 이동에이전트 시스템에서 보다 안전한 에이전트의 수행을 보장하기 위해서는 이후 설명할 에이전트 수행 결과 보호 방법을 복합적으로 제공하는 것이 바람직하다.



(그림 3) Encrypted Function 의 동작

3.2 에이전트 코드의 수행 결과 보호

에이전트의 수행 결과를 보호하고자 하는 연구의 전제는 코드를 안전하게 수행하는 것이 불가

능하다는 가정이다[8]. 따라서 이에 관한 연구는 한 호스트에서 수행을 마친 이동에이전트에 저장된 데이터가 다른 호스트에 의하여 불법으로 위, 변조되는 것을 막기 위한 방향으로 진행되어 왔다. 이들 연구의 공통적인 특징은, 각각의 호스트에서 수행된 부분 결과들 사이에 연관성을 부여한다는 것이다. 이 절에서는 두 가지 대표적인 연구 예로, 해쉬함수를 이용하여 데이터 체인을 형성하는 것과 일회용키로 수행결과를 암호화하는 방법을 설명한다.

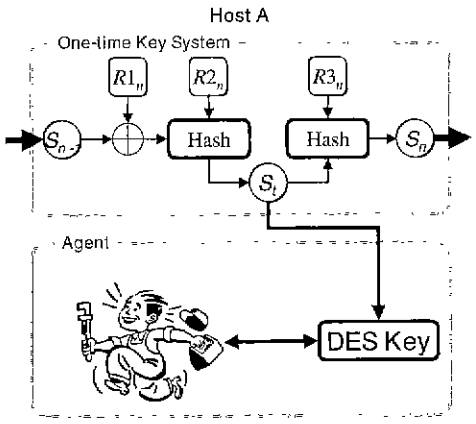
- 해쉬 체인을 이용한 데이터 무결성 보호: 이 방법은 에이전트가 각각의 호스트를 지날 때마다 수행 결과에 해쉬함수를 적용, 이동 경로 상의 호스트에서 수행한 작업 결과들 사이에 연관성을 부여하는 방법이다. 각 호스트에서의 수행 결과는 해쉬체인의 일부로 저장되며, 어떤 서버에서 에이전트 데이터가 임의로 변조 되었을 경우, 서버들이 해쉬 체인을 재 생성해 봄으로써 에이전트가 수집한 데이터가 어떤 호스트를 거치는 동안 변조되었는지를 판별해 낼 수 있다[19]. 이 방법은 에이전트 수행 결과의 무결성을 보장해 주지만, 코드와 데이터의 비밀성까지 보장해 주지는 않는다.
- 일회용 에이전트 키를 이용한 데이터 보호: 이동에이전트가 순회하는 호스트마다 각각의 키를 생성하여 데이터 암호화 키로 정의하고, 각 호스트에서의 수행 결과를 해당 호스트의 키로 암호화하여 홈 에이전트 서버로 전송하는 방법이다. 이 때, 각 호스트에서 사용하는 키들은 단방향 해쉬함수에 의해 생성되어 연관성이 부여된다. 이 방법은 데이터의 비밀성과 무결성을 제공하나, 각 서버에서 키를 생성해야 하기 때문에 전체 성능에 부담을 준다는 단점이 있으며, 에이전트가 수행 결과를 암호화하기 전에 호스트가 임의로 데이터를 변조할 경우, 적절한 대응책이 될 수 없다[15].

4. 호스트 보호를 위한 보안 모델

이동에이전트 서버가 에이전트에게 위해를 가할 수 있는 것과 마찬가지로, 이동에이전트 또한 서버에게 보안상의 위협을 가할 수 있다. 이 때의 에이전트는 악의를 가진 공격자에 의해 생성되어 시스템에 위협을 가할 수도 있는 반면, 단순한 동작 오류로 인하여 시스템의 무결성, 비밀성 및 가용성에 위협을 가할 수도 있다. 따라서, 에이전트로부터 호스트를 보호하기 위해서는 이동에이전트 유입시, 에이전트의 식별 및 인증 정보에 의거하여 한정된 권한을 부여하는 한편, 자원 접근시 시스템에 위해가 되는 연산을 수행하고 있는 것은 아닌가를 지속적으로 모니터링 해야한다. 호스트 보호를 위하여 사용되는 대표적인 방법에는 접근 제어 방법과 코드 검증 방법의 두가지 방법이 존재한다.

4.1 접근제어

접근 제어는 호스트 보안을 위해 가장 널리 쓰이는 방법이다. 현존하는 이동에이전트 시스템 중, 기본적인 형태의 접근 통제 모델을 제안하고 있는 시스템으로는 IBM Japan의 Aglet[11], Kaiserslautern 대학의 Ara[9], Dartmouth 대학의 Agent-Tcl, Minnesota 대학의 Ajanta[J] 등을 들 수 있다. 호스트 보호의 측면에서 서버에 유입되는 이동에이전트를 접근 주체, 서버가 제공하는 서비스 및 시스템 자원을 접근 객체로 규정할 수 있는데, 접근 주체인 에이전트 서버는 이동에이전트가 유입될 때 에이전트의 식별자, 생성자, 생성 서버 등의 정보를 참조하여 권한을 부여하고, 연산 수행을 제어하게 된다. 서버가 에이전트의 자원 접근을 통제하고자 할 때, 사용하는 메커니즘을 세분화하



(그림 4) 일회용 에이전트 키

면 다음과 같이 분류할 수 있다.

- 보안 관리자를 이용하는 방법 : 현재 이동에이전트 시스템의 설계 언어로 가장 널리 쓰이고 있는 언어는 자바이다. 자바의 경우 언어 수준 보안 메커니즘 중의 하나로 보안 관리자를 제공하고 있는데, 보안 관리자는 객체에의 접근 연산이 수행되는 시점에서 해당 연산이 보안 담당자가 정한 보안 규칙에 따르는지의 여부를 확인하여 그 허용여부를 결정짓는 도구이다. 보안 관리자는 객체의 연산 단위로 제공되므로 어플리케이션의 형태에 상관없이 일관된 보안 규칙의 적용을 보장할 수 있다는 장점을 제공한다. 반면, 어플리케이션 수행 시점에서 시스템이 다양한 보안 결정 정보를 반영, 유연한 접근 통제를 실시해야 하는 경우에는 보안 규칙의 설계가 매우 복잡하고 어려워진다.
- 대리자를 이용하는 방법 : 몇몇 이동에이전트 시스템에서는 이동에이전트가 서버로 유입될 때, 수행을 위해 각 이동에이전트별 대리자 객체를 제공한다. 대리자는 일반적으로 분산 컴퓨팅 환경의 동적 객체를 대표하는 객체를 의미하는데[18], 이동에이전트 시스템에서의 대리자는 에이전트의 신분과 권한을 대표하는 동시에 에이전트와 서버 간 상호작용을 매개하는 역할을 맡게 된다. 이동에이전트 서버는 이동에이전트가 수행되는 동안 에이전트의 대리자를 제공하고 보안 관련 결정을 위임, 에이전트의 자원 접근을 모니터링 하도록 할 수 있다. 에이전트는 서버가 제공하는 자원에 접근하고자 할 때 대리자에게 사용 요구를 제출하게 되는데, 이 때, 대리자는 접근 통제 리스트 정보와 이동에이전트의 식별, 인증 및 권한 정보, 접근 행위의 속성 등을 참조하여 접근이 허용가능한지의 여부를 판단한다. 대리자를 이용하는 방법의 어려움은 다양한 형태로 정의되는 접근 주체 각각에 합당한 대리자 객체를 제공하기 어렵다는 것에 있다.
- 보호막을 이용하는 방법 : 보호막이란 접근 객체, 즉 시스템 자원을 개념적으로 감싸주는 일종의 인터페이스이다. 보호막은 자신이 보호하고 있는 객체로의 접근 시도가 인지되면, 접근 통제 리스트, 접근 주체 정보 등을 참조하여 객체에의 접근이 허용 가능한지의 여부를 결정한다. 이는, 앞서 언급한 대리자를 이용하는 방법이 접근 요구를 제출하는 시점에서 권한 검사를 행하던 것과는 상대적인 개념이라고 할 수 있다. 보호막을 이용하는 방법은 접근 객체별 보호막을 정의하기만 하면, 투명한 접근 통제 메커니즘을 제공할 수 있다는 장점은 있으나 모든 접근 주체에게 동일한 접근 규칙을 적용하게 되므로 유연성이 떨어진다는 단점이 있다.

4.2 에이전트 코드의 검증

접근제어의 경우 접근 주체와 접근 개체의 인증 및 식별 정보, 접근 통제 규칙을 기반으로 연산 수행의 허용 여부를 판별하던 것에 반하여, 에이전트 코드의 검증 방법은 에이전트 코드 자체

를 분석하여 행동양식을 파악하고자 하는 방법이다. 자바 언어가 제공하는 또 다른 언어수준 보안 메커니즘인 바이트 코드 검증기[20]나 상태검사 방법이[7] 이 범주에 해당된다. 접근 통제의 경우, 접근 권한 획득에 필요한 자격을 구비한 에이전트가 일단 권한 검사 과정을 통과하고 난 후에는, 에이전트의 자원 접근 연산이 항상 올바르게 수행될 것이라고 보장하기 어렵다는 단점이 있으나, 코드 검증 방법의 경우, 주체의 코드 자체의 행동양식을 분석해 내는 방법이므로 그러한 단점을 극복할 수 있다. 그러나, 실제로 코드 분석 방법을 통해, 에이전트가 수행하고자 하는 일련의 연산에 악의성이 포함되어 있는지의 여부를 판별해 내는 것은 그다지 쉬운 일이 아니다.

5. 결 론

이제까지, 이동에이전트의 보안 문제 해결을 위하여 다양한 보안 모델과 보호 방법들이 제안, 연구되어 왔으나 아직까지는 연구의 수준이 초기 단계에 머물러 있으며, 실제 응용을 고려하여 보안 모델을 제안하고 구현한 시스템 역시 그리 많지 않은 실정이다. 호스트로부터 이동코드를 보호하는 방법에 관한 실용적인 해답 또한 아직까지 제시되지 않고 있다. 이동에이전트 시스템과 같은 응용 계층에서 독립적으로 이동코드를 보호하는 일이 쉽지 않을 뿐더러, 보호 문제에 관한 부분적인 해답이 개발된다 하더라도 overhead등의 문제점으로 인하여 이를 실용화하기 위해 비용을 감수해야함을 생각해 볼 때, 앞으로의 연구에서는 운용 시스템을 설계시점에서부터 이동 코드의 보호 문제를 고려하여, 운용 시스템과 이동에이전트 시스템 계층에서의 보안 역할 분담 및 협력을 통해 이동 코드의 안전한 수행을 보장하고자 하는 연구가 진행되어야 할 것이다.

끝으로, 본자료의 일부는 한국 과학재단의 특정기초연구(98-0102-11-01-3)의 과제 수행 결과를 포함하고 있음을 밝힌다.

참고문헌

[1] C. G. Harrison, D. M. Chess, and A. Kershenbaum, "Mobile Agents: Are they a good idea?", Technical report, IBM T.J. Watson Research Center, 1995.

[2] D. Kotz and R. Gray, "Mobile Code: The Future of the Internet", Mobile Agents in the Context of Competition and Co-operation (MAC3) (Seattle, Washington, USA, May 1 1999), May 1999.

[3] J. E. White, "Mobile Agents", Software Agents, MIT Press and American Association for Artificial Intelligence, 1997.

[4] F. Hohl, "Time Limited Blackbox Security: Protecting Mobile Agents From Malicious Hosts", Mobile Agents and Security, Springer-Verlag, pp. 99-113, 1998.

[5] T. Sander and Chr. Tschudin, "Towards Mobile Cryptography", the 1998 IEEE Symposium on Security and Privacy.

[6] F. Hohl and K. Rothermel, "A Protocol Preventing Blackbox Tests of Mobile Agents", ITG/VDE Fachtagung Kommunikation in Verteilten Systemen (KiVS'99), Springer, 1999.

[7] W. Farmer, J. Gurrman, and V. Swarup, "Security for mobile agents: Authentication and state appraisal", to appear in the proceedings of the European Symposium on Research in Computer Security(ESORICS), Lecture Notes in Computer Science, September 1996.

[8] W. Farmer, J. Guttman, and V. Swarup, "Security for mobile agents: Issues and require-

ments", to appear In National Information Systems Security Conference, National Institute of Standards and Technology, October 1996.

[9] H. Peine, "Security Concepts and Implementation in the Ara Mobile Agent System", 7th IEEE Workshop on Enabling Technologies: Infrastruc- ture for Collaborative Enterprises, Stanford Univer- sity, USA, June 1998.

[10] J. Baumann, F. Hohl, K. Rothermel, and M. Strasser, "Mole - Concepts of a mobile Agent System", The World Wide Web Journal, speciall issue on Software Agents, 1998.

[11] G. Karjoth, D. B. Lange, and M. Oshima, "A Security Model for Aglets", IEEE Internet Computing, Vol. 1, No. 4, pp. 68-77, July - August 1997.

[12] R. Gray, "Agent Tcl: A flexible and secure mobile agent system", In Proceedings of the Fourth Annual Tcl/Tk Workshop, Monterey, Cal., pp. 9-23, July 1996.

[13] Ananda Tripathi and Neeran Karnik, "Protected Resource Access for Mobile Agent-based Distributed Computing", the 1998 ICPP Workshops on , pp. 144 -153, 1998.

[14] Jusung Baek, DongIk Lee, "A design of a protocol for detecting a mobile agent clone and its correctness proof using Coloured Petri Nets", Technical Report TR-DIC-CSL-1998-002, Info.&Comm., K-JIST, 1998.

[15] Jongyoul Park, DongIk Lee, "Data Protection in Mobile Agents", submit to the European Symposium on Research in Computer Security (ESORICS) 2000.

[16] R. S. Sandu, E. J. Coyne, "Role-Based Access Control Models", IEEE Computer, Vol. 29, No. 2, pp. 38-47, February 1996.

[17] J. Y. Levy and J. K. Ousterhout, "A Safe Tcl Toolkit for Electronic Meeting Places". In Proceedings of the First USENIX Workshop on Electronic Commerce, pages 133-135, July 1995.

[18] M. Sharpiro, "Structure and Encapsulation in Distributed Systems: The Proxy Principle", In Proceedings of the 6th International Conference on Distributed Computing Systems, pages 198-204. IEEE, 1986.

[19] G. Karjoth, N. Asok an, c. Gulcu, "Protecting the Computation Results of Free Roaming Agents", Second International Workshop MA'98, Stuttgart, Germany, September 1998.

[20] L. Gong, "Java Security Architecture (JDK1.2)", Technical Report, JavaSoft, July 1998, Revision 0.5

박 종 열

1996년 충남대학교 컴퓨터공학과 졸업
1999년 광주과학기술원 정보통신공학과 석사
1999년-현재 광주과학기술원 정보통신공학과 박사과정

관심분야 : 보안 시스템, 에이전트 시스템, 분산 시스템, Petri net 이론 등

신 욱

1998년 동국대학교 컴퓨터 공학과 졸업
2000년 광주과학기술원 정보통신공학과 석사
1999년-현재 광주과학기술원 정보통신공학과 박사과정

관심분야 : 접근통제 시스템, 이동에이전트 시스템, 워크플로우 시스템, Coloured Petri-net 등

이 동 익

1985년 영남대학교 전기공학과 졸업
1989년 Osaka Univ. 전자공학과 석사
1993년 Osaka Univ. 전자공학과 박사
1990년-1995년 Osaka Univ. 전자공학과 research associate
1993년-1994년 Univ. of Illinois at Urbana-Champaign, visiting assistant professor
1995년-현재 광주과학기술원 정보통신공학과 부교수

관심분야 : 에이전트 시스템 및 보안, 비동기회로 설계/CAD. Petri net 이론 등