ID기반 디지털 다중 서명 기술을 적용한 안전한 이동 에이전트 시스템의 설계

유 성 진[†]·김 성 열[††]·이 옥 빈[†]·정 일 용[†††]

요 약

이동에이전트 시스템은 기하급수적으로 증가하는 분산처리 환경과 이동컴퓨팅에 기여할 수 있다는 점으로 인해 주목받고 있는 기술이지만, 심각한 보안문제를 안고 있다. 본 연구는 이동 에이전트 시스템이 가질 수 있는 보안 공격을 NIST 문서에 기초하여 분석하였다. 이들 공격으로부터 이동 에이전트 시스템을 보호하기 위해서 ID에 기반한 키(key) 분배 기법과 디지털 다중 서명(Digital Multi-signature)기법을 이용하여 이동 에이전트 시스템을 위한 보안 프로토콜을 제안한다. NIST에서 제안한 이들 문제를 해결하기 위해서는 이동 에이전트 보안과 에이전트 플랫폼 보안을 수행하여야 한다. 기존 프로토콜은 둘 중의 하나만을 언급한 반면에, 제안된 프로토콜은 이들 두 가지 측면 모두에서 보안을 만족시키고자 하였다. 제안된 프로토콜은 1)키관리의 단순화, 2)보안 서비스 만족(기밀성, 무결성, 인증, 부인방지), 3)생명성 보장, 4)실행결과데이터 보호, 5)재전송공격방지 등의 보안 특성을 만족시킨다. 또한 에이전트 실행의 각 단계를 매 시스템마다 검증함으로써 메시지의 변경시에 곧바로 탐지한다.

A Design of Secure Mobile Agent Systems Employing ID based Digital Multi-Signature Scheme

Seong-Jin Yoo[†]·Seong-Yeol Kim[††]·Ok-Bin Lee[†]·Il-Yong Chung[†††]

ABSTRACT

Mobile agent system comes into the spotlight since it contributes largely to mobile computing on distributed network environment. However, this system has a number of significant security problems. In this paper, we analyze security attacks to mobile agent system presented by NIST[3]. In order to protect this system from them, we suggest a security protocol for mobile agent system by employing ID based key distribution and digital multi-signature scheme. To solve these problems described in NIST, securities for mobile agent and agent platform should be accomplished. Comparing with other protocols, our protocol performs both of these securities, while other protocols mentioned only one of them. Proposed protocol satisfies simplicity of key management, providing security service such as confidentiality, integrity, authentication and preventing reputation, liveness guarantee, protection of excution-result data and preventing replay attack. Furthermore, it is designed to detect message modification immediately by verifying each step of agent execution at a corresponding server.

키워드 : 이동 에이전트(Mobile Agent), 디지털 다중서명 기법(Digital multi-signature scheme), ID기반 키 분산(ID-based key distribution)

1. 서 론

이동 에이전트(Mobile Agent)는 소프트웨어 에이전트 기술에 이동 코드 개념을 결합한 새로운 이동 컴퓨팅 기술을 의미한다. 이동 에이전트는 기하급수적으로 증가하는 분산처리 환경과 이동 컴퓨팅의 변화 때문에 주목받고 있는 기술이다[1]. 그러나 이동 에이전트 시스템은 분산 어플리케이션의 구성에 유연한 환경을 제공하는 반면, 해결하여야할 보안 문제를 안고 있다[2].

이동 에이전트의 공격으로부터 호스트 컴퓨터를 보호하는 문제는 에이전트의 인증과 접근 제어로써 해결될 수 있지만 에이전트 서버의 공격으로부터 에이전트를 보호하는 문제는 일반화된 보호방법이 없어 현재 해결하기 어려운 문제로 남아 있다[3].

에이전트는 에이전트 자신을 실행하는 환경을 제공하는 서버에게 모든 것이 노출되므로 서버는 에이전트의 코드나 상태를 변경 또는 삭제하여 에이전트의 행동을 방해할 수 있기 때문이다. 이러한 특성으로 인하여 에이전트는 에이전트 서버의 불법적인 공격에 매우 취약하다. 이러한 공격을 탐지하고 예방하기 위해서는 에이전트의 모든 실행 상태를 에이전트 생성자에게 전달하고 생성자가 이를 확인할 수 있는 방법이 필요하다. 그러므로 다양한 응용에서 안전하게 이동 에이전트 시스템을 적용하기 위해서는 보다 효과적으로 에이전트 코드와 상태를 보호하는 방법에 대한 연구가

† 준 회 원 : 조선대학교 대학원 전자계산학과
†† 정 회 원 : 울산과학대학 컴퓨터정보학부 교수
††† 종신회원 : 조선대학교 컴퓨터공학부 교수
논문접수 : 2002년 4월 18일, 심사완료 : 2002년 12월 12일

필요하다. 또한 이동 에이전트 시스템의 보안 문제는 이동 에이전트의 실행 결과 데이터를 적절히 보호하는 방법과 이동 에이전트의 생명성을 보장하는 문제가 존재한다. 따라서 이동 에이전트 시스템 보안 문제는 이동 에이전트 시스템이 해결하여야 할 중요한 연구 과제로 부상하고 있다.

정적인 에이전트의 보안 문제는 전통적인 보안 대책으로 접근이 가능하지만, 이동 에이전트 시스템에 적용할 경우 이동 에이전트의 특징인 이동성을 파괴할 수 있다. 에이전트는 신뢰할 수 있는 영역(domain) 외부에서 자율적으로 운행되고 동작 할 수 있어야 하지만, 특정 플랫폼에 대한 정보가 잘 알려지지 않은 경우에 에이전트를 보호하기 어렵다. 즉 에이전트가 플랫폼 사이에서 이동할 때, 에이전트를 받아들이는 플랫폼은 유입된 에이전트에 의해 어떤 간섭이 발생할지 알 수 없으며, 이동 에이전트는 해당 플랫폼의 악성 여부를 결정할 수 없다. 따라서 이동 에이전트 시스템은 새로운 형태의 보안 메커니즘이 필요하다.

본 논문에서는 NIST 문서인 [3]을 기반으로 하여 이동 에이전트의 보안 위협요소를 기술하고 이를 해결하기 위한 프로토콜을 제안한다. 제안된 프로토콜은 ID 기반의 디지털 다중 서명 기술을 이용하고 있다. ID를 이용한 암호시스템은 ID를 공개키로 사용하기 때문에 공개키를 인증할 필요성이 없어지고, 공개키 디렉토리를 유지하지 않아도 되는 장점이 있다. 또한 디지털 다중 서명(Digital Multi-signature)기법은 메시지 인증 기능과 사용자 인증 기능을 같이 수행한다는 장점이 있다.

본 논문의 구성은 다음과 같다. 2장에서는 이동 에이전트 시스템의 보안 위협 사항을 분석하고, 기존 연구에 대하여 살펴보았다. 3장에서는 프로토콜 제안을 위한 기반 기술에 대하여 살펴보고 4장에서는 이동 에이전트 보안 시스템 프로토콜을 제안하고 5장에서 제안한 프로토콜을 분석하였다. 그리고 6장에서는 본 논문의 결론을 도출한다.

2. 보안 위협과 기존연구

2.1 이동 에이전트 시스템의 보안 위협

이동 에이전트 시스템의 보안 공격은 크게 정보 노출, 서비스 거부, 정보 손상으로 구분할 수 있다. 에이전트 시스템에는 많은 구성 요소가 있지만 보안 공격을 고려하는데 에이전트와 에이전트 플랫폼이 주요한 요소가 되는데 에이전트 플랫폼은 호스트에 에이전트 서버가 탑재된 것으로 에이전트 실행 환경을 제공한다.

[3]은 이동 에이전트 시스템의 공격을 에이전트의 에이전트 플랫폼 공격, 에이전트 플랫폼의 에이전트공격, 에이전트의 에이전트 공격, 또 다른 요소의 에이전트 시스템 공격으로 구분하고 있다.

(1) 에이전트의 에이전트 플랫폼 공격(Agent-to-Platform)

악성 이동 에이전트에 의한 플랫폼의 공격은 불법적 자원의 사용, 플랫폼 내부의 정보 누출, 자원의 고갈, 데이터 삭제, 플랫폼의 파괴, 플랫폼의 서비스 봉쇄, 서비스 거부 공격 등을 고려할 수 있다.

(2) 에이전트의 에이전트 공격(Agent-to-Agent)

이동 에이전트에 대한 악성 이동 에이전트의 공격은 에이전트가 다른 에이전트의 약점을 탐색하거나 다른 에이전트를 직접 공격하는 방법으로 에이전트 플랫폼 내부에서 이동 에이전트가 가진 정보를 탈취 또는 에이전트의 정상적인 활동을 방해하기 위하여 악성 에이전트가 공격을 시도할 수 있다.

(3) 에이전트 플랫폼의 에이전트 공격(Platform-to-Agent)

이 공격은 호스트의 실행 환경이 특정 에이전트를 공격하여 에이전트가 위치하고 있는 컴퓨터 시스템 내에서는 에이전트에게 부여된 역할을 할 수 없게 만드는 것이다. 개방된 이동 에이전트 시스템에서 이동 에이전트는 일반적으로 자신의 플랫폼이 아닌 다른 플랫폼 위에서 작업을 수행하게 된다. 이런 환경에서 이동 에이전트의 동작에 중대한 문제가 발생할 수 있다. 악성 플랫폼이란 다른 영역에 포함되어 있는 에이전트에게 실행이 가능한 환경을 제공하여, 에이전트를 공격을 시도하는 플랫폼으로 정의할 수 있다. 악성 에이전트 플랫폼이 시도할 수 있는 공격은 에이전트의 코드 추적, 에이전트가 가진 데이터 추적, 제어 흐름의 추적, 에이전트 코드의 변경, 데이터의 변경, 제어 흐름의 변경, 코드의 부정확한 실행, 특정 호스트로 위장, 실행 거부, 다른 에이전트와의 상호 작용 추적, 다른 에이전트와의 상호 작용 변경, 에이전트가 플랫폼에 요청한 결과를 조작하여 전송하는 경우 등을 고려할 수 있다.

(4) 다른 요소의 에이전트 시스템 공격(Other-to-Agent Platform)

이동 에이전트와 플랫폼 모두에 대한 다른 개체의 공격은 에이전트와 플랫폼이 제공하는 특정 서비스 봉쇄를 목적으로 하여 특정 플랫폼에 대한 에이전트 생성, 또는 이동 에이전트의 동작을 모두 봉쇄하기 위한 공격 유형으로 나누어 볼 수 있다.

위의 공격 가능 부분별 위협 요소를 정리하면 <표 1>과 같다.

〈표 1〉 공격 가능 부분별 위협요소

분 류	(1)	(2)	(3)	(4)
위장(Masquerading)	○	○	○	○
서비스 거부(Denial of Service)	○	○	○	○
허가되지 않은 접근(Unauthorized Access)	○	○		○
부인(Repudiation)		○		
침해(Eavesdropping)			○	
변경(Alteration)			○	
복제 및 재전송(Copy and Replay)				○

○ : 위협요소 존재

2.2 기존 연구

이동에이전트 시스템의 보호와 관련한 연구는 이동에이전트 시스템의 주요 구성요소인 이동에이전트 플랫폼 보호와 이동에이전트 보호로 나누어 살펴볼 수 있다.

2.2.1 에이전트 플랫폼 보호

이동 에이전트 플랫폼 보호를 위한 연구들은 네트워크 자원을 남용하거나 실행 시간이 많이 걸리고 비용이 높다는 등의 지적을 받고 있다. [4]에서는 암호화 기법을 이용하여 에이전트의 실행 상태를 검증자에게 전달하고, 검증자는 실행 상태에 대한 추적을 통해 에이전트를 보호하는 구조를 제안하였으나 이는 네트워크 자원을 많이 사용하게 되어 많은 비용을 요구하고 실행 코드가 커진다는 단점이 있다. 또한 [5]에서는 증명 검사 방법을 제안하고 있으나 이 또한 실용적인 응용에는 적합하지 않은 것으로 평가된다. [6]에서는 단순 디지털 서명 반복하여 에이전트를 이동시키고 최종적으로 감사 도구를 이용하여 에이전트 여행 정보와 실행 상태 정보에 보안 시스템을 사용함으로써 에이전트를 보호하고 있다. 그러나 이 방식은 에이전트의 실행 도중에 발생하는 경유 정보에 대한 변경 행위만을 감사할 수 있고, 실행 결과 데이터의 변조에 대하여는 최종적으로 감사할 수 있다. 변경 행위를 즉각 발견하지 못함으로써 불필요한 오버헤드를 가질 수 있으며 단순 서명 기법을 반복 사용함으로써 서명 길이가 길어지는 단점이 있다.

2.2.2 이동 에이전트 보호

이동 에이전트의 보호를 위해 제안된 방법으로는 공격자에게 코드를 분석할 충분한 시간을 주지 않도록 하는 접근방법[7], 에이전트를 실행하기 위한 환경을 제공하는 완전한 컴퓨터 TPE(Tamper Proof Environment)를 두고 에이전트를 암호화하는 접근방식[8], 서버의 시간과 환경에 의해 키를 생성해 암호화된 에이전트를 실행할 수 있도록 하는 접근 방식[9], 이동 에이전트가 가지고 있는 이동성에 따른 위험성 제거와 코드의 중요도에 따라 코드를 분할 및 재생성 하는 접근방식[10] 등이 제안되었다. 또한 에이전트에 대한 서버의 위협은 무시하고 불법적인 에이전트로부터 서버를 위한 보안 프로토콜[11]이 제안되어 있다. 그러나 이러한 시스템들은 이동에이전트와 서버(에이전트 플랫폼) 간의 상호 인증을 제공하지 않는다는 단점이 있다.

3. ID 기반 보안 기술과 디지털 다중 서명

3.1 ID 기반 보안 기술

ID를 이용한 암호시스템에서는 공개키에 해당하는 ID와 ID에 대응되는 비밀키가 있으며, 비밀키를 만들어 주는 키 발급 센터가 존재한다. 키 발급 센터만이 비밀키를 생성할 수 있는 능력을 갖고 있으므로 키 발급 센터이외에는 누구도 ID를 변경할 수 없다. 이 방식은 shamir가 1984년 기본 개념을 처음 제안하였다[12]. ID를 이용한 암호 시스템에서는 ID를 공개키로 사용하기 때문에 공개키를 인증할 필요가 없다. 따라서 공개 기록집을 작성하지 않아도 되는 장점이 있다.

ID를 이용한 암호 시스템은 ID를 이용한 개인 식별(identity based identification scheme), ID를 이용한 서명(identity based signature scheme), ID를 이용한 키분배(identity based key distribution scheme) 등으로 나누어진다.

3.2 디지털 다중 서명 기법

한사람이 메시지에 전자적으로 서명하는 것으로써 단순 서명(single signature)이라 한다. 이와 같은 전자 서명 기술은 전자 문서 교환을 위해 중요한 역할을 수행한다. 단순 서명 방식을 직접 반복함으로써 다중 서명에 적용할 수 있으나 문서의 길이가 증가하고, 검증하기 위해서는 서명자의 수만큼 검증 과정을 거치기 때문에 서명자의 수가 많은 경우에 서명 속도 등에서 비효율적이다. 따라서 효율적인 디지털 다중 서명 방식이 필요하다. 동일한 메시지에 대해 여러 사람이 전자적으로 서명하는 것을 '디지털 다중 서명'이라 한다. 디지털 다중 서명 방식이 갖추어야할 기본적인 조건들은 서명문 길이의 고정, 검증 가능성, 부정 조기 검출성, 비밀 유지성, 공통성 등이다[13]. 이러한 조건들을 만족하며 단순 서명 방법의 비효율적인 문제를 해결하기 위한 다중 서명 방법은 두 개의 큰 소수와 서명자의 직위에 따라 서명이 이루어지는 Itakura-Nakamura 방법[14], 작은 소수의 곱을 이용하여 RSA 방법을 확대 적용한 Okamoto 방법[15], RSA 방식과 같은 전단사 공개키 암호 시스템과 단방향 함수로 이루어진 Brickell-Lee-Yacobi 방법[16], Fiat-Shamir 방식에 근거하여 만들어진 Ohta-Okamoto 방법[17] 등이 있다.

3.3 새로운 프로토콜에 대한 접근

본 논문에서 제안하고자하는 이동에이전트 시스템의 보안 구조는 에이전트 플랫폼과 이동에이전트의 보호라는 두 가지 측면에서 2장의 위협요소들에 대해 안전한 구조를 제시하고자 한다. 제안된 보안 프로토콜은 Fiat-Shamir의 ID기반 디지털 다중 서명법을 기초로 설계되었다. Fiat-Shamir의 ID기반의 디지털 서명은 이산 대수 문제에 근거하여 암호학적인 안전성이 있으며 이산 대수 문제는 암호, 인증 방식을 구성하기 위한 대표적인 문제이다. 제안된 디지털 서명을 이용한 보안 프로토콜의 안전성은 Fiat-Shamir의 방식과 동일하다. 또한 Fiat-Shamir 방식에 근거하고 있기 때문에 서명속도가 RSA에 근거한 방식보다 20배 정도 빠르고 ID에 근거한 서명 방식으로 공개키 디렉토리가 불필요하며 키관리를 단순화할 수 있다. 따라서 공개키 방식에서 수반되는 PKI 등이 불필요하다. 또한 서명순서의 제약을 받지 않으며 서명 메시지의 길이는 보안레벨에 의해 결정된다. 따라서 이동 에이전트 시스템의 구성시 좀 더 유연한 구조를 가질 수 있다.

4. 이동 에이전트 시스템 보안 프로토콜

4.1 개 요

본 논문에서 제안하는 이동 에이전트 보안 프로토콜은 Fiat-Shamir의 ID기반의 디지털 다중 서명법을 기초로 하여 에이전트가 경유하는 각각의 에이전트 플랫폼마다 서명을 수행하는 다중 서명 기능을 포함한다. 이와 같은 다중 서명을 통해 송수신 에이전트 플랫폼을 인증하고, 에이전트 플랫폼에서의 에이전트 실행을 감시하며, 에이전트 라우팅 정보의 변경 또는 삭제를 탐지할 수 있다. ID기반의 키 분배 방식을 사용하여 전송되는 정보를 보호하며, 에이전트의 실행 결과 데이터의 무결성과 기밀성을 보장할 수 있다. 본 장에서 사용되는 표기법은 <표 2>와 같다.

〈표 2〉 제안된 프로토콜의 표기법

표 기	설 명
f	공개된 일방향함수
h	공개된 강한 해시함수
AP_i	에이전트 이동 경로상의 i번째 에이전트 플랫폼의 ID
AP_H	에이전트를 생성한 에이전트 플랫폼 ; 홈(HOME)
MA	이동에이전트 ; $\{A_{name}, A_{code}, A_{route}, exe\_results_i, A_{sign}\}$
A_{name}	에이전트의 이름 (owner + URN)
exe_result_i	AP_i가 에이전트를 실행하여 얻은 실행결과 데이타
$\log_i$	$exe_results_i$를 ABS와의 공유키로 암호화한 데이타
$exe\_results_i$	AP_i 가 생성한 실행결과 데이터의 연접
A_{code}	에이전트의 실행코드
A_{sign}	에이전트의 실행결과에 대한 다중서명
H_H	AP_H의 이동 에이전트 코드 값
AP_{route}	홈이 선택하는 경로의 집합(이동경로)
ABS	Agent Base Server로 AP에게 키를 분배하는 기관이며 에이전트실행결과를 최종 점검한다.
$k_{n,m}$	n과 m의 공유키
$E_k(M)$	메시지 M이 키 k에 의하여 암호화됨
R_n	AP_n이 생성한 랜덤 수
$R_{n,m}$	AP_n이 AP_n에게 전송하기 위하여 생성한 랜덤 수
$t1_n$	AP_n이 AP_{n+1}와 통신시 replay 공격에 대비한 timestamp
$t2_n$	AP_n이 실행결과를 생성한 시간의 time stamp
S_i	식별자 i의 비밀키
S_{ij}	식별자 i,j의 해시함수 값의 역수의 제곱근(서명정보 생성요소)
e_{ik}	i의 서명정보 비트열 중 k번째 요소
Y_i	AP_i의 서명 정보

4.2 ABS의 서비스 제공 프로토콜

ABS는 AP등록, 삭제 프로세스를 수행하고, 각 에이전트 플랫폼으로부터 요청되는 공유키 생성을 위한 프로세스와 홈으로부터 받은 에이전트 코드와 이동 경로 정보를 바탕으로 에이전트를 감시할 프로세스를 수행하게 된다. 또한 4.4절과 같이 에이전트 실행을 감사하고 이동 에이전트 실행결과를 홈에 제공하게 된다.

4.2.1 이동 에이전트 서버 등록

이동에이전트 서비스를 수행하기 위한 에이전트 플랫폼은 서비스를 수행하기 이전에 에이전트 서버로서의 등록이 이루어져야 한다. 서버 등록을 위해서 에이전트 플랫폼이 ABS에 등록을 요청하면 ABS는 각각의 에이전트 플랫폼 고유의 키를 생성하고 배포한다. 이 후에 이동에이전트 서버 프로그램을 제공함으로써 이동 에이전트 서비스를 제공할 수 있는 환경을 갖추도록 한다.

[등록요청] AP_i는 에이전트 서버로서 역할을 수행하기 위하여 ABS에 등록을 요청한다. 이때 AP_i의 AP 등록 요청 메시지에는 자신의 식별 정보를 포함해야한다.

[키 발생] ABS는 다음과 같은 절차에 의해 키를 발생시킨다.

① 정수의 소인수분해의 어려움에 근거한 두 개의 큰 소수 p와 q를 선택하여 $N = p \times q$를 계산한다.
② Euler 함수값 $\varphi(N) = (p-1)(q-1)$와 서로소인 e를 구하고 이에 근거한 d를 구한다.

$$\gcd(e, \varphi(N)) = 1 \quad (1)$$
$$ed = 1 \bmod \varphi(N) \quad (2)$$

③ $GF(p)$와 $GF(q)$에 포함되는 원시근 g를 구한다.
④ AP_i를 등록하고 각 AP_i에 대하여 S_i, $S_{ij}(1 \le j \le k)$와 k_{ik}를 구한다.

$$S_i = AP_i{}^d \bmod N \quad (3)$$
$$I_{ij} = f(AP_i, j), \quad j = 1, 2, \cdots, k \quad (4)$$
$$I_{ij}{}^{-1} = S_{ij}{}^2 \bmod N, \quad (\exists I_{ij},\ I_{ij} \in QR_N)^{1)} \quad (5)$$
$$k_{ik}, \quad I_{ik}{}^{-1} \in QR_N \quad (6)$$

⑤ AP_i에 대하여 발생한 키를 저장한다

(그림 1) ABS의 키 발생 과정[1)]

[키 배포] ABS는 발생한 키에 대해 AP들을 물리적으로 식별한 후 p, q를 비밀리에 보관하고 식 (7)을 스마트 카드에 담아 발급 배포한다.

$$(N, e, g, S_i, f, h, S_{i1}, \cdots, S_{ik}, k_{i1}, \cdots, k_{ik}, AP_1, \cdots, AP_m, ABS) \quad (7)$$

$k_{i1}, \cdots, k_{ik}$는 $I_{ij}{}^{-1}$이 이차잉여류인 j를 구한 것이다.

차후에 변경 사항이 발생하는 경우는 AP_i와 ABS간의

1) QR_N : modulus N에 대하여 이차 잉여류인 집합 전체를 의미한다.

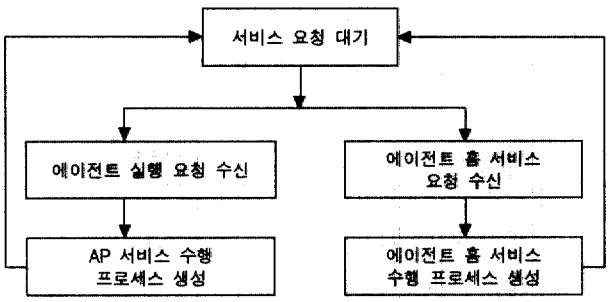
세션 공유키를 이용하여 변경 정보를 전달받고 스마트 카드 키를 변경한다.

4.2.2 에이전트 감시 프로세스

에이전트 감시 프로세스는 에이전트 플랫폼들이 요청해 오는 과정에 따라 공유키를 생성하는 절차와 에이전트를 수신하고 에이전트를 검증하는 절차로 구성되며, 에이전트 수행 과정에서 공유키 생성에 참여하지 않는 에이전트 플랫폼이 있을 경우 발견하고 조치할 수 있도록 진행된다. 또한 에이전트 플랫폼이 에이전트 실행 전 서명을 검증하는 과정에서 적절하지 못한 불법 수행이 있는 것으로 판단되면 *ABS*에 보고하도록 설계되었는데 이 정보를 감시 프로세스가 받아 처리하도록 한다.

4.3 에이전트 플랫폼의 서비스 시나리오

에이전트 플랫폼은 에이전트 실행 요청이나 에이전트 서비스 요청을 받음에 따라 적절한 프로세스를 생성하게 된다. 여기에서 에이전트 실행 요청이란 이동 경로에 따라 이동해 온 이동에이전트의 실행 요구를 의미하며, 에이전트 홈 서비스 요청이란 사용자가 에이전트를 생성하고자 요구하는 경우를 의미한다. (그림 2)는 이러한 에이전트 플랫폼의 서비스 시나리오를 나타낸 것이다.



(그림 2) 에이전트 플랫폼의 서비스 시나리오

4.3.1 에이전트의 생성(creagteAgent)

홈(AP_H)에서 에이전트를 생성하는 경우 *ABS*에 이를 알리고 *ABS*로 하여금 생성된 에이전트의 보안관리를 수행토록 한다. 이를 위한 절차는 다음과 같이 설계되었다.

① 에이전트 생성시 다음과 같이 에이전트 이동 경로를 생성한다.

$$A_{route} = AP_1 \| AP_2 \| \cdots \| AP_n \tag{8}$$

② 에이전트의 실행 데이터 값과 다중서명 값이 null인 상태에서 다음을 계산한다.

$$A_{sign} = H_H = f(A_{code}), \quad A_{code}, A_{sign} \in MA \tag{9}$$

③ (그림 6)과 같이 *ABS*와의 세션 공유키 세션 절차를 수행하여 $k_{H,ABS}$를 생성한다.

④ 최초로 에이전트를 실행할 AP_1와 세션 공유키 세션 절차를 수행하여 AP_1와의 공유키 $k_{H,1}$를 생성한다.

⑤ 다음과 같이 에이전트 코드와 경로 정보를 *ABS*에 전송한다.

$$AP_H \rightarrow ABS \ : \ E_{k_{H,ABS}}(MA, t1_H) \tag{10}$$

⑥ 데이터를 다음과 같이 서명한다.

$$X_H = R_H{}^2 \bmod N \tag{11}$$

$$(e_{H1}, \cdots, e_{Hk}) = h(A_{route}, exe\_results_H, H_H, X_H) \tag{12}$$

$$Y_H = R_H \prod_{e_{Hj}=1} S_{Hj} \bmod N, \quad j = 1, 2, \cdots, k \tag{13}$$

$$A_{sign} = \{H_H, X_H, k_H, Y_H\}, \quad A_{sign} \in MA \tag{14}$$

⑦ AP_H는 최초로 에이전트를 실행할 AP_1에게 다음을 전송한다.

$$AP_H \rightarrow AP_1 \ : \ E_{k_{H,1}}(MA, t1_H) \tag{15}$$

4.3.2 에이전트 실행(executeAgent)

에이전트가 AP_i로 이동하면 호스트는 에이전트를 하나의 쓰레드로 처리한다.

① AP는 이동 에이전트를 실행하여 실행 결과 데이터에 의해 다음과 같이 $\log_i$를 만들고 *exe_results*을 갱신한다.

$$\log_i = E_{k_{i,ABS}}(exe\_result_i, t2_i) \tag{16}$$

$$exe\_results_i = exe\_results_{i-1} \| \log_i \tag{17}$$

이 절차에 의하여 에이전트 실행 결과 데이터는 다른 에이전트 플랫폼으로부터 보호될 수 있고 ABS에 의해 검증이 가능하다.

② 실행결과에 대해 다음과 같이 서명한다.

$$R_i \in Z_N$$

$$X_i = R_i{}^2 X_{i-1} \bmod N, \quad i = 1, 2, \cdots k \tag{18}$$

$$(e_{i1}, \cdots, e_{ik}) = h(A_{route}, exe\_results_i, X_i, H_H) \tag{19}$$

$$Y_i = Y_{i-1} R_i \prod_{e_{ij}=1} S_{ij} \bmod N, \quad j = 1, 2, \cdots, k \tag{20}$$

$$A_{sign} = \{H_H, k_H, X_H, k_1, k_2, \cdots, k_i, X_1, X_2, \cdots, X_i, Y_i\}, \tag{21}$$

$$k_i = \{k_{i1}, k_{i2}, \cdots, k_{ik}\}, \ k_{ij}, I_{ik} \in QR_N$$

(그림 3) 에이전트 실행 후 서명절차

4.3.3 에이전트의 이주(TransferAgent)

AP_H에서 에이전트를 전송하거나 AP_i에서 에이전트의 이주요구가 발생하여 AP_{i+1}로 에이전트를 이동시킬 때 AP_i와 AP_{i+1}간에 TransferAgent 절차를 수행하게 된다. Transfer-Agent 절차 수행의 성공은 지금까지의 에이전트 수행에 보

안상의 아무런 문제가 발생하지 않았음을 보장하도록 설계되었다.

① AP_i와 AP_{i+1}간의 세션 키 $AP_{i,i+1}$를 생성한다.

② 에이전트를 수신할 AP_{i+1}는 ABS와의 세션키 $k_{i+1,ABS}$를 생성한다. 이 과정을 통해 ABS는 AP_{i+1}이 에이전트를 수신하는 단계에 있음을 알 수 있다. 세션키는 서명검증 후 식 (16) 단계에서 사용된다.

③ AP_i는 다음과 같이 에이전트코드와 함께 서명된 실행결과를 전송한다.

$$AP_i \rightarrow AP_{i+1} : E_{k_{i,i+1}}(MA, t1_i) \quad (22)$$

④ AP_{i+1}은 다음과 같이 이전 단계의 서명을 검증한다.

$$I_{xj} = f(AP_x, j), \quad x = 1,2,\cdots,i, \ j \in k_i \quad (23)$$

$$(e_{i1},\cdots,e_{ik}) = h\,(A_{route}, exe\_results_i, X_i, H_{it}) \quad (24)$$

$$X_i = Y_i^2 \prod_{x=1}^{i} \prod_{e_{xj}=1} I_{xj} \bmod N, \ j \in k_i \quad (25)$$

식 (25)가 성립하면 유효한 결과임을 알 수 있다.

(그림 4) 에이전트 플랫폼의 서명 검증 과정

⑤ 만일 검증 결과가 올바르지 않으면 AP_{i+1}는 즉시 ABS에 보고하고 그렇지 않으면 에이전트를 수행한다.

(그림 5)는 구성 요소들간의 상호 동작을 나타낸 것이다.

4.4 에이전트 실행 감사(auditAgent) 및 결과 제공

ABS는 에이전트 생성시에 수행되었던 $AP_H \rightarrow ABS$: $E_{k_{H,ABS}}(MA, t2_H)$에 의해 에이전트의 이동계획을 알고 있다. 그리고 에이전트가 이동할 때마다 AP_i와 세션키를 공유하게 되므로, ABS는 에이전트가 활동을 중지하는 경우 이를 검출할 수 있다.

에이전트가 정상적으로 수행되어 예정된 경로를 순회하여 ABS에 도달하면 (그림 4)와 같이 서명의 전과정을 검증할 수 있다.

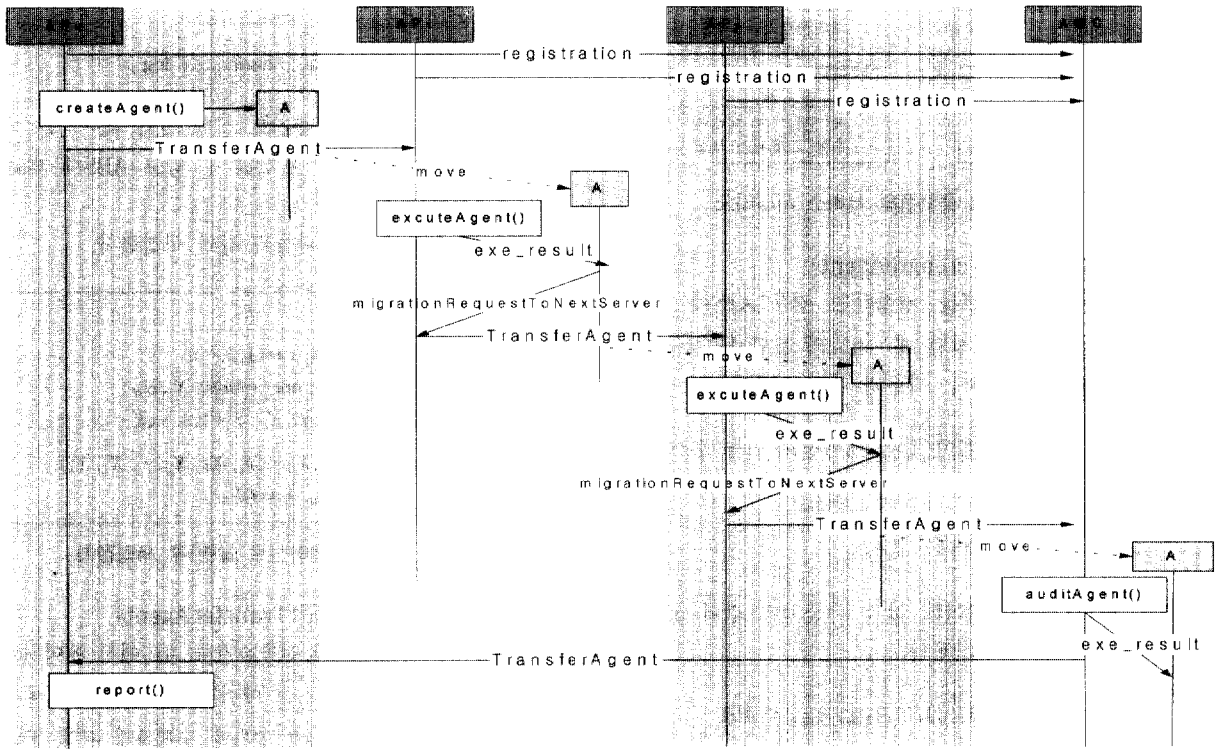
서명이 유효하면 각 AP_i들과 설정하였던 세션(session) 공유키 테이블 <표 3>을 참조하여 각 서버에서 t시간에의 실행 결과를 복호화하여 획득할 수 있다. 이 결과를 홈에 암호화하여 전달함으로써 이동에이전트의 실행을 마치게 된다.

〈표 3〉 ABS의 세션 공유키 테이블

에이전트 서버의 ID	세션 공유키
AP_1	$g^{e \cdot R_1 \cdot R_{ABS}}$
AP_2	$g^{e \cdot R_2 \cdot R_{ABS}}$
⋮	⋮
AP_{n-1}	$g^{e \cdot R_{n-1} \cdot R_{ABS}}$
AP_n	$g^{e \cdot R_n \cdot R_{ABS}}$

4.5 세션(session) 공유키 생성 절차

등록을 마친 AP는 에이전트 플랫폼으로 서비스할 수 있다.



(그림 5) 수행 요소들 간의 상호 동작

AP가 다른 여러 AP와 연관되어 이동 에이전트 서비스를 수행함에 신뢰성 있는 암호 통신 방법이 필요하다. 이를 위해 두 AP간에 세션 공유키를 생성하는 절차를 거친다. 이 세션 공유키는 ID에 기반한 대칭적 비밀키로서 AP_i는 ABS, AP_{i-1}, AP_{i+1}와의 사이에 세션 공유키가 필요하게 된다. 이 방법은 공개키 방식과 다르게 세션마다 키 교환이 이루어지지만 공개키 방식이 가지는 키 디렉토리 침해에 의한 문제 및 신뢰할 수 있는 제 3자의 불신 문제를 해결할 수 있다. 세션 공유키를 생성하는 절차는 다음과 같다.

[R_i 생성] AP_i는 랜덤 수 $R_i \in Z_N$ (Z_N : {0,1,…,N−1})을 선택한다.

[C_{AP_i} 계산] AP_i는 식 (26)를 계산하여 AP_{i+1}에 전송한다.

$$C_i = S_i \cdot g^{R_i} \bmod N \tag{26}$$

[$C_{AP_{i+1}}$ 계산] AP_{i+1}는 랜덤 수 R_{i+1}를 선택한 후 식 (27)을 계산하여 AP_i에게 전송한다.

$$C_{i+1} = S_{i+1} \cdot g^{R_{i+1}} \bmod N \tag{27}$$

[키 생성 k_i] AP_i는 식 (28)과 같이 AP_{i+1}와의 공유키 k_i를 계산한다.

$$\begin{aligned} k_i &= (C_{i+1}{}^e / AP_{i+1})^{R_i} \bmod N \\ &= g^{e \cdot R_i \cdot R_{i+1}} \bmod N \end{aligned} \tag{28}$$

[키 생성 k_{i+1}] AP_{i+1}는 다음과 같이 AP_i와의 공유키 k_{i+1}를 계산한다.

$$\begin{aligned} k_{i,i+1} = k_{i+1} = k_i &= (C_i^e / AP_i)^{R_{i+1}} \bmod N \\ &= g^{e \cdot R_i \cdot R_{i+1}} \bmod N \end{aligned} \tag{29}$$

(그림 6) 세션 공유키 생성 절차

5. 제안된 프로토콜의 보안 서비스 분석

본 논문에서 제안하는 이동 에이전트 시스템 보안 프로토콜은 ID 방식을 기반으로 키를 분배하고 서명하고 정보를 암호화하여 이동 에이전트 시스템의 보안 위협과 전통적 보안 문제를 해결할 수 있도록 설계되었다. 이 프로토콜은 이동 에이전트와 플랫폼간에 상호 인증 가능하고 에이전트 코드와 경로는 해당 플랫폼에 공개되나 변경할 수 없으며, 에이전트 실행 데이터는 공개되지 않는다. 또한 에이전트의 생명성을 보장할 수 있는 구조이므로 시스템의 fault-tolerance가 가능하다. 또한 에이전트가 경유하는 각각의 에이전트 플랫폼마다 서명을 수행하는 다중 서명 기능을 포함하여 송수신 에이전트 플랫폼을 인증하고, 기밀성, 무결성, 부인 방지 등이 만족된다. 그리고 재전송 공격에 방어하기 위하여 $t1_i$를 정보 전송시 사용하도록 설계되었다. 본 절에서는 제안된 프로토콜이 이동 에이전트 시스템에 대한 보안 위협에 대처할 수 있음을 설명한다.

5.1 이동 에이전트 보안

5.1.1 에이전트 서버의 인증

에이전트 서버의 사칭 위협을 방지하기 위해서는 이동 에이전트를 생성한 서버와 이동할 서버 사이에서 서로의 컴퓨터가 서로 상대방의 identity를 확인할 수 있어야 한다. 제안된 시스템은 trusted center인 ABS를 두어 서비스를 제공하고자 하는 AP에게 제공한다. 상호 통신하는 각각의 AP는 ABS로부터 상대방의 ID를 확인하고, ABS에서 분배하는 공통의 e, g, N을 사용하여 전송 정보를 서명/암호화하므로 ABS로부터 인증된 AP와 통신하게 됨을 알 수 있다.

5.1.2 에이전트 코드와 데이터의 기밀성

제안된 시스템은 전송되는 데이터의 기밀성을 보장하기 위하여 ID에 기반한 암호 키 분배 방식을 사용하고 있다. 이 방식은 공개키 방식의 단점인 디렉토리 관리의 문제점 등을 극복할 수 있을 뿐 아니라 세션마다 다른 키를 사용할 수 있다는 장점을 가지고 있다. 공유키를 생성한 AP_i와 AP_{i+1}, ABS는 정보를 $AP_i \rightarrow AP_{i+1} : E_{k_{i,i+1}}(X)$와 같이 전송함으로써 제삼자에게 데이터가 누설되는 것을 방지할 수 있다. 또한 AP_i는 식 (16)과 같이 실행 상태 정보를 암호화함으로써 ABS 이외에 어떤 AP들에게도 자신이 작성한 데이터를 노출시키지 않는 특징이 있다.

5.1.3 에이전트 코드와 데이터의 무결성

이동 에이전트의 코드와 데이터가 네트워크를 통해 전송되는 중에 제삼자에 의해 변형되지 않았음을 증명할 수 있는 방법이 필요하다. 제안된 시스템은 (그림 3)과 같이 에이전트 코드, 경로 정보, 상태 정보에 대한 메시지 인증코드를 생성하여 $(A, AP_{route}, statustable, (e_{11}, \cdots, e_{1k}), Y_1)$와 같이 전송하기 때문에 각 AP는 에이전트 실행전 검증 과정을 통해 에이전트 코드, 경로 정보, 상태 정보에 대한 무결성을 확인할 수 있고 문제가 있는 것으로 판단될 경우 즉각 ABS에게 보고할 수 있다.

5.1.4 실행 결과 데이터의 보호

보안 위협 분석에서 에이전트 실행 데이터는 항상 변화하므로 이에 대한 적절한 보호 대책이 제안되어 있지 않고 있다[3]고 하였다. 그러나 본 논문에서는 에이전트 플랫폼이 에이전트의 실행 데이터를 읽거나 불법적으로 수정함으로써 에이전트의 행동에 영향을 줄 수 있는 문제점을 개선하였다. 에이전트를 실행하게 되는 에이전트 플랫폼은 ABS와 공유키를 생성하여 실행 결과 데이터를 암호화함으로써 다른 에이전트 플랫폼들에게 이 정보를 알 수 없도록 할 수 있다.

5.1.5 에이전트의 생명성 보장

에이전트는 여행 중에 악의적인 에이전트 플랫폼에 삭제/종료되거나 네트워크상의 오류로 인하여 분실되는 문제가

발생할 수 있다. 그러므로 에이전트의 실행 및 여행 과정을 감시하여 원래의 목적된 경로를 지나 의도했던 작업을 실행할 수 있도록 하기 위한 장치가 필요하다. 제안된 시스템은 에이전트 라우팅 정보와 이를 통한 *ABS*의 에이전트 여행 과정의 감시 기능에 의해 에이전트의 생명성을 보장할 수 있다. 즉, *ABS*는 에이전트 홈으로부터 전달된 에이전트의 이동 경로에 의해 에이전트 플랫폼들에 공유키 생성 요청을 검사하는 방법으로 에이전트의 실행 및 여행 과정을 감시할 수 있고 문제가 발생한 경우 적절한 조치를 함으로써 에이전트의 생명성을 보장할 수 있다.

5.1.6 에이전트 전송의 부인 방지

제안된 시스템은 에이전트를 보내는 *AP*가 에이전트에 디지털 다중 서명 기법에 기초하여 디지털 서명하여 보냄으로서 부인 방지를 처리할 수 있다. AP_n은 서명 정보를 생성하기 때문에 부인하였을 경우 검증 과정을 통해 근거를 제시할 수 있다.

5.1.7 이동 에이전트 실행 감사

이동 에이전트는 코드와 데이터로 구성되어 있는데 코드는 생성된 후 *AP*를 이동하는 도중에 변하지 않지만 실행 데이터는 항상 변화한다. 심각한 위협은 *AP*가 에이전트의 실행 데이터를 불법적으로 수정할 때 발생할 수 있다. 즉, 호스트 컴퓨터들이 이동하면서 얻은 정보들이 한 호스트에서 모두 삭제되거나 불법적으로 수정되어 에이전트의 행동에 영향을 줄 수 있다. 이를 탐지하기 위하여 에이전트가 이동한 호스트들에 대한 이동 경로의 확인과 각 호스트들에서 실행된 에이전트의 실행 데이터에 대한 로그 정보를 바탕으로 한 감사 기능이 제공되어야 한다. 에이전트의 이동 경로는 각 호스트들이 에이전트의 결과를 부인할 수 없도록 구성되어야 한다. 그리고 에이전트의 실행 데이터에 대한 로그 정보는 각 호스트들이 디지털 서명하도록 함으로써 정보 제공에 대하여 부인을 할 수 없도록 구성되어야 한다. 제안된 시스템은 식 (16)~식 (17)과 같이 실행 결과 데이터가 암호화되어 $\log_i$에 저장되고 $exe_results_i$이 서명 기록에 포함되므로 다음 *AP*가 검사할 수 있다. 또한 *ABS*는 홈으로부터 제공받은 AP_{route}를 토대로 공유키 형성에 참여하기 않은 *AP*가 에이전트 불법 종료 등의 조작을 취할 가능성이 있음을 감시할 수 있다. 즉, 공유키 형성이 되었더라도 다음 *AP*가 공유키 형성에 동작을 취하지 않으면 마지막 공유키를 형성한 *AP*가 에이전트를 종료하였음을 알아낼 수 있다.

5.2 에이전트 플랫폼 보안

5.2.1 에이전트의 인증

에이전트 서버는 이동 에이전트를 받으면 이 에이전트가 신뢰할 만한 서버에서 생성된 이동 에이전트인지를 확인하여 에이전트에게 서버에 대한 접근 레벨을 부여한다. 에이전트는 에이전트를 생성한 측에 의해 디지털 서명되거나 에이전트 생성자가 믿을 수 있는 인증기관에게 요청해 인증될 수 있다. 제안된 시스템에서 에이전트는 에이전트를 생성한 측에 의해 디지털 서명된다. *AP*는 이동 에이전트를 받으면 이 에이전트가 신뢰할만한 *AP*에서 생성된 이동 에이전트인지를 검증한다. 이를 통해 에이전트를 인증하게 되고 에이전트에게 *AP*에 대한 접근 레벨을 부여한다.

5.2.2 에이전트의 호스트 컴퓨터에 대한 리소스 접근 제어

이동 에이전트는 호스트 컴퓨터의 에이전트 시스템에 의해 실행되는데 에이전트는 호스트 컴퓨터의 자원을 접근하는 것을 제어한다. 호스트 컴퓨터에 대한 접근 제어로써 에이전트의 호스트에 대한 불법적인 행동, 호스트 컴퓨터의 인가되지 않은 파일 접근을 탐지하고 막을 수 있다. 에이전트 서버는 이동 에이전트 수행에 필요한 자원을 할당하기 위하여, 이동 에이전트를 수신하기 전에 에이전트의 식별자, 작업을 의뢰한 사용자, 에이전트를 생성한 시스템의 식별자를 요구하여 에이전트의 권한을 검사한다. 에이전트 서버에서 에이전트의 활동이 시작되면, 에이전트 서버는 접근 통제 메커니즘을 행하여 에이전트의 상태를 불법적으로 변경하지 못하도록 한다. 제안된 프로토콜을 적용한 시스템은 적정한 리소스 접근제어 정책을 적용하는데 제한을 받지 않는다.

5.3 제안된 프로토콜의 특징

이동 에이전트 시스템의 보안을 위하여 제안된 프로토콜의 주요 기능을 공격자에게 코드를 분석할 충분한 시간을 주지 않도록 한 Hohl 방식[7], 에이전트를 실행하기 위한 TPE(Tamper Proof Environment)를 두고 에이전트를 암호화하는 Whlhelm-Stamann 방식[8], 서버의 시간과 환경에 의해 키를 생성해 암호화된 에이전트를 실행할 수 있도록 한 Riordan-Schneier 방식[9], 디지털 서명과 감사 도구를 이용한 Back[6] 방식들과 비교하면 <표 4>와 같다.

제안된 프로토콜은 암호화 기술을 기반으로 하는 보안 기술을 이용하여 제안된 프로토콜로써, 다중 서명 기법을 이용하여 Back 방식의 서명 길이가 길어진다는 단점을 해결하였으며, 양방향 인증, 기밀성, 무결성, 부인 방지를 만족시키고 있다. 또한 에이전트에 대한 불법적인 조작이 발생한 경우에 대한 조치를 수행하는 fault-tolerance가 고려되었고, 실행 결과 데이터를 에이전트 플랫폼들에게 공개되지 않도록 하는 기능을 가지고 있다. 이는 다른 시스템들에서 지원하지 못하고 있는 fault-tolerance, 실행 결과 데이터 보호 등의 문제 해결책을 제안함으로써 이동 에이전트 시스템의 실제 응용에 사용될 수 있도록 하였다.

〈표 4〉 제안된 프로토콜의 특징 비교

항 목	Hohl 방식	Wilhelm-Stamann 방식	Riordan-Schneier 방식	Back 방식	제안된 프로토콜
보호 방식	시간제한 코드혼합	하드웨어	환경키	정보은닉 단순서명 방법	정보은닉 다중서명 기법
암호화 방식	비대칭	비대칭	비대칭	공개키에 의한 비대칭키	ID 키에 의한 대칭키
에이전트 이동방식	1pass	1pass	1pass	1pass	1pass
인증 방식	단방향	단방향	단방향	단방향	양방향
경로 공개	공개	공개	공개	공개	공개
fault-tolerance	×	×	×	△(부분허용)	○
정보 기밀성	○	○	○	제안되어 있지 않으나 가능	○
정보 무결성	×	×	×	제안되어 있지 않으나 가능	○
부인방지	×	×	×	○	○
실행결과 데이터 비공개	공개	공개	공개	공개	비공개

6. 결 론

본 논문은 이동 에이전트 시스템이 안고 있는 보안 문제를 해결하고자 하였다. 제안된 프로토콜은 이동 에이전트 및 에이전트 시스템 보안 위협에 대처하기 위하여 ID를 이용한 키 분배 기법과 Fiat-Shamir 디지털 서명 방식에 기초한 다중 서명 방법을 이용하여 에이전트와 에이전트 플랫폼의 양방향 인증, 실행 결과 데이터의 보호, 생명성 보장을 함께 처리하였으며 중간 검증이 가능하도록 제안되어 불필요한 오버헤드를 갖지 않도록 하였다. ID기반의 암호화 방식은 키 분배, 디지털 서명 등에 적용될 수 있으며 키 관리의 단순화라는 장점이 있으며 공개키 방식에 비하여 서명 처리 속도가 빠르다는 장점을 가지고 있다. 제안된 이동 에이전트 보안 프로토콜의 특징을 요약하면 다음과 같다.

첫째, 키 관리를 단순화시킨 구조이다. 제안된 시스템은 공개키 방식의 단점인 디렉토리 관리의 문제점 등을 극복할 수 있을 뿐 아니라 암호화를 위한 세션 키 구성을 위한 별도의 방법을 두지 않아도 된다. 또한, 각각의 세션마다 서로 다른 키를 사용할 수 있는 장점을 가진다.

둘째, 기밀성, 무결성, 인증, 부인 방지를 보장한다. 제안된 시스템에서는 실행 결과 정보를 암호화함으로써 *ABS*이외에 어떤 *AP*들에게도 자신이 작성한 데이터를 노출시키지 않는다. 각 *AP*는 에이전트를 실행하기 전에 검증과정을 통해 에이전트 코드, 경로 정보, 상태 정보에 대한 무결성을 확인할 수 있고 문제가 있는 것으로 판단될 경우 즉시 *ABS*에게 보고할 수 있도록 제안되었다. 또한 디지털 서명을 사용하여 부인방지를 처리할 수 있다. 제안된 시스템에서 에이전트는 에이전트를 생성한 측에 의해 디지털 서명된다. *AP*는 이동 에이전트를 받으면 이 에이전트가 신뢰할만한 *AP*에서 생성된 이동 에이전트인지를 검증하도록 제안되었다.

셋째, 에이전트의 생명성을 보장하는 기능을 가지고 있다. 정보를 수신한 *AP*는 에이전트를 실행하고 실행 결과 데이터를 암호화하기 위하여 *ABS*에 공유키 생성을 요청하여야 한다. 이를 이용해 *ABS*는 *AP*가 에이전트 불법 종료 등의 조작을 취할 가능성이 있음을 감시할 수 있도록 함으로써 생명성을 보장할 수 있도록 하였다.

넷째, 실행 결과 데이터를 보호한다. 본 연구에서는 *AP*가 에이전트의 실행데이터를 읽거나 불법적으로 수정함으로써 에이전트의 행동에 영향을 줄 수 있는 문제점을 개선하였다.

다섯째, 암호화 정보 안에는 타임스탬프를 사용하여 재전송 공격에 대처하도록 제안되었다.

제안된 이동 에이전트 보안 모델은 현재 컴퓨팅 기술에서 이슈가 되고 있는 이동 에이전트 시스템의 보안 문제 해결에 대한 가능성을 디지털 다중 서명 기법을 통해 제시하였으며 실제 이동 에이전트 시스템의 구성에 적용될 수 있다. 향후 연구는 이동 에이전트 서버들의 디렉토리 서비스와 어려운 문제로 지적되고 있는 자율 경로 배정에 적합한 이동 에이전트 시스템의 보안 구조에 대한 연구가 계속적으로 필요하다.

참 고 문 헌

[1] Dale, J. and Mamdani, E., "Open Standards for Interoperating Agent-Based Systems," Software FOCUS, Wiley, 2001.

[2] Poslad, S. and Calisti, M., "Towards Improved Trust and Security in FIPA Agent Platforms," Autonomous Agents 2000 Workshop on Deception, Fraud and Trust in Agent Societies, Spain, 2000.

[3] Jansen, W. and Karygiannis, T., "Mobile Agent Security," NIST Special Publication 800-19, 1998.

[4] Vigna, G., "Protecting Mobile Agents through Tracing," In 3rd ECOOP Workshop on Mobile Object Systems, Jyväskylä, Finland, Proceedings. Lecture Notes In Computer Science 1241 Springer, 1997.

[5] Bennet, S., "A Sanctuary for Mobile Agents," DARPA Workshop on Foundations for Secure Mobile Code Workshop, pp.26-28, 1997.

[6] 백주성, 이동익, "디지털 서명과 감사 도구(Audit trail)를 이용한 이동 에이전트의 보호", 한국정보과학회 학술발표논문집, 제24권 제2호, 1997.

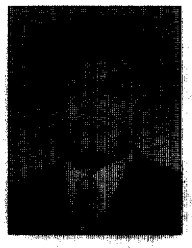
[7] Hohl, F., "An approach to solve the problem of malicious hosts," Universitat Stuttart, Fakultat informatik, Fakultatsbericht Nr., 1997.

[8] Wilhelm, U. and Stamann, S. and Buttyan, L., "Protecting the Itinerary of Mobile Agents," Workshop on Mobile Object Systems, Proceedings. Lecture Notes In Computer Science, Vol.1543, Springer, 1998.

[9] Riordan, J. and Schneier, B., "Environmental Key Generation towards Clueless Agents," Mobile Agents and Security, pp.15-24, Springer-Verlag, 1998.
[10] 박재경, 원유헌, "안전한 이동 에이전트 게이트웨이의 설계 및 구현", 한국정보과학회논문지C, 제8권 제2호, pp.240-249, 2002.
[11] Ordille, J., "When agents roam, who can you trust?," Proc. of the First Conference on Emerging Technologies and Applications in Communications, Porland, 1996.
[12] Shamir, A., "Identity-based cryptosystem and signature scheme," Proc. of CRYPTO 84, pp.47-57, Springer-Verlag, 1985.
[13] 김승주, 원동호, "특수 디지틀 서명방식에 대한 고찰", 통신정보보호학회지, 제6권, pp.30-31, 1996.
[14] K. Itakura and K. Nakamura, "A public-key Cryptosystem Suitable for Digital Multisignature," NEC J. RES, Dev.71, pp.1-8, 1983.
[15] T. Okamoto, "A Digital Multisignature Scheme Using Bijective Public-key Cryptosystems," ACM Trans. on Comp. systems, Vol.6, No.8, pp.432-441, 1988.
[16] E. F. Brickell, P. J. Lee and Y. Yacobi, "Secure Audio Teleconference," Advances in Cryptology-Cryto '87, Lecture Notes in Computer Science 293, pp.418-426, 1988.
[17] K. Ohta and T. Okamoto, "A Digital Multisignature Scheme Based on the Fiat-Shamir Scheme," Proceedings of Asiacrypt '91, pp.75-79, 1991.

유 성 진

e-mail : white@mail.chosun.ac.kr
1998년 조선대학교 전자계산학과(학사)
2000년 조선대학교 대학원 전자계산학과 (이학석사)
2000년~현재 조선대학교 대학원 전자계산학과(박사 3학기)
관심분야 : 정보보안, 분산처리, 전자상거래, 멀티미디어



김 성 열

e-mail : kimsy@ulsan-c.ac.kr
1994년 조선대학교 전자계산학과(학사)
1996년 조선대학교 대학원 전자계산학과 (이학석사)
2000년 조선대학교 대학원 전자계산학과 (이학박사)
2002년~현재 울산과학대학 컴퓨터정보학부 전임강사
관심분야 : 정보보안, 전자상거래

이 옥 빈

e-mail : lobin@hyun.chosun.ac.kr
1990년 조선대학교 전자계산학과(학사)
1993년 조선대학교 대학원 전자계산학과 (이학석사)
1995년~1997년 충북대학교 대학원 전자계산학과(박사 수료)
관심분야 : 시스템 이론, 컴퓨터 네트워크, 정보보안

정 일 용

e-mail : iyc@mina.chosun.ac.kr
1987년~1991년 City University of New York in U.S.A 전산학박사
1979년~1983년 한양대학교 공과대학 공학사
1999년~2000년 정보전산원장
1997년~1999년 정보과학대학 학장보
1991년~1994년 한국전자통신연구소 선임연구원
1993년~현재 조선대학교 전자정보공과대학 컴퓨터공학부 교수
관심분야 : 네트워크 보안, 전자상거래, 분산시스템 관리, 코딩 이론, 병렬 알고리즘