이동 에이전트 수행 결과에 대한 부정 검출 프로토콜

김 희 연[†] · 신 정 화[†] · 신 원[††] · 이 경 현[†††]

요 약

이동 에이전트 시스템은 기존 클라이언트-서버 모델이 가진 제한 사항을 해결하고, 네트워크 환경의 보다 효율적인 활용을 가능하게 하는 방안 중 하나로 제안되었다. 이동 에이전트 시스템은 네트워크 상의 이질적인 호스트 환경에서 원활히 동작할 수 있는 인터페이스를 제공하는 시스템이며, 신뢰성있는 실행을 위하여 이동 에이전트를 구현하는 이동 코드에서의 보안 문제 해결이 선행되어야 한다. 본 논문에서는 이동 에이전트가 에이전트 소유자의 요구에 따라 네트워크 상을 이동하여 실행될 때, 각각의 호스트에서 수행된 결과 값에 서명 기법과 해쉬 체인 기법을 적용하여 전방향 무결성, 부인 방지, 부정 검출을 가능하게 하는 프로토콜을 제안한다.

A Forgery detection protocol for protection of mobile agent execution results

Hee-Yeon Kim[†] · Jung-Hwa Shin[†] · Weon Shin[††] · Kyung-Hyune Rhee[†††]

ABSTRACT

Mobile agent systems offer a new paradigm for distributed computation and a one of solution for limitation of existent Client-server model. Mobile agent systems provide interface that can migrate from host to host in a heterogenous network. For secure execution, it must solve security problem of mobile code before. In this paper, we are propose the protocol that applied signature technique and hash chain technique. This protocol enable one to offer forward integrity, non-repudiation, and forgery detection, when mobile agents are perform the task by migrating a network.

키워드 : 이동 에이전트(mobile agent), 부정 검출(forgery detection)

1. 서 론

네트워크 환경에서 가장 일반적으로 사용되고 있는 클라이언트-서버 모델은 높은 대역폭 요구, 전송 지연, 클라이언트와 서버간의 지속적인 상호 작용 요구에 의한 과부하의 발생 등 여러 가지 문제점에 직면하고 있다[1]. 클라이언트-서버 모델이 가지는 이러한 문제점들을 해결하기 위한 방안 중 하나로 등장한 이동 에이전트 시스템은 새로운 분산 처리 패러다임으로써 기존의 클라이언트-서버 모델을 대체할 수 있을 것으로 기대되며 많은 연구가 진행되고 있다[2].

이동 에이전트는 사용자를 대신하여 사용자의 요구사항을 자율성과 이동성을 가지고 처리하는 프로그램으로 이동 코드(Mobile Code) 기술에 기반을 두고 있으며, 네트워크를 통해 전달될 때 현재 객체의 상태(Status)와 실행 가능한 코드(Executable Codes), 수행 데이터(Execution Data)를 가지고 직접 호스트 사이를 이동하여 실행된다[3, 4]. 또한, 가상 머신 상에서 수행되기 때문에 다양하고 이질적인 환경에서도 유연한 대처가 가능하며 독립적인 여러 에이전트가 하나의 작업을 수행하기 위해 통신하고 협력함으로써 분산처리의 효율성을 높일 수 있다. 그러므로, 이동 에이전트 시스템은 네트워크로 연결된 독립적인 다수의 에이전트가 네트워크 상의 이질적인 호스트 환경에서 원활히 동작할 수 있는 인터페이스를 제공하는 시스템으로 정의할 수 있다.

그러나, 이동 에이전트가 가지는 대표적인 특징인 이동성(Mobility)은 보안 측면에서 몇 가지 문제점을 가지기 때문에 이를 보완할 필요가 있다. 이동성은 Java, ActiveX, Telescript, Agent-Tcl 등과 같이 네트워크를 통해 원격지로 전송되어 실행되는 이동 코드 기술로 구현되며, 이동 코드 기술은 이질적인 시스템에서의 융통성 있는 수행과 효율적인 네트워크 자원 사용을 가능하게 하는 반면, 원격지 시스템에서 실행되기 때문에 이동 코드의 출처에 대한 인증, 이동

※ 이 논문은 2001학년도 두뇌한국21 사업에 의하여 지원되었음.
† 준 회 원 : 부경대학교 대학원 전자계산학과
†† 준 회 원 : 안철수 연구소
††† 종신회원 : 부경대학교 전자컴퓨터정보통신 공학부 교수
논문접수 : 2002년 7월 30일, 심사완료 : 2002년 10월 26일

코드 자체에 대한 무결성 검증, 이동 코드가 수행되는 시스템에 대한 인증, 이동 코드에 대한 접근 제어 등의 문제가 해결되어야 한다. 이동 코드 기술은 이동 에이전트 시스템을 구현하기 위한 핵심적인 기술로써 이동 에이전트 시스템의 보안 문제를 해결하기 위해서는 이동 코드의 보안 문제의 해결이 반드시 선행되어야 한다[5, 6].

따라서, 본 논문에서는 안전한 이동 에이전트 시스템을 구현하기 위한 선행 단계로써 이동 코드로 설계되는 에이전트의 실행 결과에 대하여 전방향 무결성(forward integrity), 부인 방지, 부정 검출을 보장할 수 있는 프로토콜을 제안하고자 한다.

2장에서 시스템의 취약성과 이동 에이전트의 수행 결과를 보호하기 위한 기존 방안을 살펴보고, 3장에서 이동 에이전트의 수행 결과 보호를 위한 새로운 프로토콜을 제안하고, 4장에서 결론을 맺는다.

2. 관련 연구

2.1 이동 에이전트 시스템의 취약성

이동 에이전트는 인터넷과 같은 공개 네트워크 상의 호스트들을 직접 이전해 다니면서 주어진 작업을 수행하기 때문에, 에이전트의 상태, 실행 가능한 코드, 이전 호스트로부터 얻은 데이터를 공격자로부터 안전하게 보호하는 것이 중요 관심사이다. 그러나, 현재 대부분의 공개 네트워크는 안전하지 않은 것으로 알려져 있고 이동 에이전트가 실행되는 호스트 또한 반드시 정직하다는 것을 보장할 수 없기 때문에 이동 에이전트에 대한 공격에 대해 분석한 후 적절한 대응이 필요하다[6-8].

이동 에이전트에 대한 공격은 다음과 같이 분류된다.

- 도청(Eavesdropping) : 에이전트를 수행하는 호스트는 에이전트의 제어 흐름, 코드, 상태, 데이터와 에이전트의 통신 등에 접근이 가능하기 때문에 에이전트가 가지는 중요 정보를 도청할 수 있다.
- 위장(Masquerade) : 악의적인 목적을 가지는 호스트가 정당한 호스트인 것처럼 위장하여 에이전트를 수행하고 에이전트로부터 데이터와 상태를 전달받을 수 있다.
- 부인(Repudiation) : 이동 에이전트를 수행했던 호스트는 자신의 내부에서 실행되었던 에이전트의 수행 결과를 부인할 수 있다.
- 불법적인 접근 및 수정(Unauthorized Access and modification) : 에이전트는 호스트 내부에서 수행되기 때문에 호스트는 에이전트가 저장하고 있는 에이전트의 수행 상태와 데이터는 물론 에이전트의 실행 코드에 대해 불법적으로 접근하고 접근한 에이전트의 수행 상태, 데이터, 실행 코드에 대한 임의 변경, 삭제가 가능하다.
- 서비스 거부(Denial of Service) : 호스트는 에이전트가 접근하고자 하는 시스템 자원과 서비스에 대해서 부분적으로 또는 완전히 방해함으로써 에이전트의 정당한 수행을 지연시키거나 막을 수 있다.

2.2 이동 에이전트 수행 결과 보호를 위한 기존 연구

이동 에이전트의 소유자는 에이전트에게 작업을 전담한 후 주어진 작업의 수행을 마치고 돌아왔을때 에이전트가 가져오는 모든 수행 결과를 신뢰할 수 있어야 한다. 그러기 위해서는 에이전트가 호스트 상에서 수행하는 작업에 대하여 변경이나 부정이 없었음을 신뢰할 수 있어야 한다. 또한, 경매 정보와 같이 다른 호스트에 의해 보여지는 것만으로 침해가 될 수 있는 데이터의 경우에는 데이터의 비밀성이 추가로 보장되어야 한다. 따라서, 에이전트에 가해질 수 있는 여러 가지 공격에 대하여 에이전트를 보호하기 위한 연구가 필수적인데 현재까지는 이를 위한 부분적인 해결 방안들이 제안되고 있다. 에이전트의 수행 결과를 보호하기 위해 제안된 방안으로는 호스트가 에이전트 수행 결과에 대해서 서명을 생성할 수 있도록 에이전트 내부에 실행 가능한 함수를 포함하는 분리할 수 없는 서명(Undetachable Signature) 방안[9], 에이전트의 수행 결과를 부분적으로 캡슐화(encapsulate)하여 에이전트 소유자만이 수행 결과 값을 검증할 수 있도록 하는 "Sliding Encryption" 방안[10], 에이전트 수행 후 결과 값을 변경할 수 없도록 해쉬 체인을 이용한 방안[11], 부정 검출과 수행 결과 보호를 동시에 수행하기 위한 암호학적 추적 방안 등이 있다.

[9]에서 제안하고 있는 분리할 수 없는 서명(Undetachable Signature) 방안은 에이전트 내에 함수 합성 기법(Function Composition Techniques)을 적용한 서명 생성 함수와 데이터를 함께 보내서 에이전트가 방문하는 호스트에서 데이터에 포함된 RSA 서명 값을 검증하여 에이전트 소유자의 신원을 확인하고 서명 생성 함수에 호스트의 입력 값을 넣어서 새로운 해쉬 값과 호스트의 서명 값을 계산할 수 있도록 하는 방안이다. 이 방안은 이동 에이전트의 소유자에 대한 신원 확인과 무결성을 제공하는 반면, 상호 인증과 비밀성은 제공하지 않는다.

[10]에서 제안하고 있는 "Sliding Encryption" 방안에서 이동 에이전트는 에이전트 소유자의 공개키를 가지고 호스트들을 방문하게 되며 방문한 각각의 호스트로부터 계산된 정보를 에이전트 소유자의 공개키를 이용하여 암호화한 후 옮긴다. 에이전트가 작업을 마치고 돌아왔을때 지금까지 얻은 암호화된 데이터에 대해서 에이전트 소유자가 자신의 개인키를 이용하여 복호함으로써 작업에 대한 결과를 얻을 수 있다. 이 방안은 호스트에서 암호화를 수행하기 전에 부가

적인 무결성 측정이 이루어져야 하며 "빠른" 암호화 보다는 이동 에이전트, 스마트 카드, 모바일 유닛 등 저장 공간이 제한된 분야에 적용 가능하도록 "저장 공간을 확보"하고자 하는 방안이다.

[11]에서 제안하고 있는 해쉬 체인을 이용한 방안은 이동 에이전트가 방문하는 호스트에서 수행되는 결과에 해쉬 함수를 적용하여 각각의 수행 결과 값 사이에 연관성을 부여하여 수정 공격이 가해졌을 경우 어떤 호스트에서 공격이 일어났는지 검출하는 것이 가능하다.

3. 이동 에이전트 수행 결과에 대한 부정 검출 프로토콜

3.1 제안 프로토콜

제안 프로토콜에서 이동 에이전트 *agent*은 에이전트 소유자(Originator) O를 출발하여 호스트 $H_i\,(1 \le i \le n)$를 방문한 후 다시 에이전트 소유자에게 돌아오는 것으로 가정한다. 이동 에이전트 *agent*는 방문하는 각각의 호스트부터 데이터 $Data_i$를 얻게 된다. 이동 에이전트 소유자 H_0의 서명을 생성하기 위하여 RSA 전자 서명 기법[12]을 사용하는 반면, 에이전트를 수행하는 네트워크 상의 각각의 호스트는 서명을 생성하기 위하여 일반적인 서명 기법을 사용한다. 논문에서 사용되는 표기법은 다음과 같다.

〈표 1〉 표기법

O	이동 에이전트 소유자(Originator)의 ID
H_i	i 번째 호스트의 ID
$Hvalue_i$	i 번째 호스트의 결과에 대한 해쉬 값
S_i	i 번째 호스트까지의 해쉬 값을 더한 O의 RSA 서명 값
M_i	i 번째 호스트의 서명 값 $M_i = Sig_{pr_{H_i}}(Data)$
pr_{H_i}	i 번째 호스트의 개인키
pu_{H_i}	i 번째 호스트의 공개키
$Response_i$	호스트에서 수행된 결과 값
$Request$	에이전트가 수행해야 하는 작업에 대한 설명

(1) 이동 에이전트는 에이전트 소유자 O를 출발하기 전에 필요한 값을 계산한다.

① $Hvalue_0 = hash(O, Request)$

에이전트 소유자는 서명하려고 하는 메시지인 자신의 ID O와 $Request$에 암호학적 해쉬 함수를 적용한 해쉬 값 $Hvalue_0$을 계산한다.

② $S_0 = Hvalue_0^{pr_O} \bmod n_O$

개인키 pr_O를 적용하여 $Hvalue_0$에 대한 에이전트 소유자의 RSA 서명 값을 생성한다.

(2) 이동 에이전트는 계산한 $((O, Request), S_0)$ 값을 가지고 O를 출발하여 다음 목적지인 호스트 H_1에 도착한다.

① 호스트 H_1은 이동 에이전트가 가지고 온 에이전트 소유자의 서명 값을 검증한다. 호스트 H_1은 서명 값에 O의 공개키 pu_O를 적용한 $S_0^{pu_O} \bmod n_O$값과 이전 호스트로부터 전달받은 평문 형태의 데이터 $(O, Request)$에 해쉬 함수를 적용하여 계산한 $Hvalue_0*$ 값이 동일한지 비교하여 동일한 값임이 판명되었을 때 O의 정당한 서명 값으로 신뢰하게 된다.

② $Hvalue_1 = hash(H_1, O, Response_1)$

호스트 H_1은 이동 에이전트를 실행하고 에이전트에서 수행을 요청한 $Request$에 대한 결과 값인 $Response_1$을 생성한다. 서명 값을 생성하기 위해 자신의 ID인 H_1, 에이전트 소유자의 ID인 O, 호스트 H_1에서 생성한 $Response_1$을 암호학적 해쉬 함수를 적용하여 해쉬 값 $Hvalue_1$을 계산한다.

③ $M_1 = Sig_{pr_{H_1}}(Hvalue_1)$

개인키 pr_{H_1}을 적용하여 $Hvalue_1$에 대한 호스트 H_1의 서명 값을 생성한다.

④ $S_1 = S_0^{Hvalue_1} \bmod n_O$

호스트 H_1은 자신의 해쉬 값을 적용한 O의 서명 값을 생성한다. O는 이 값을 통하여 에이전트의 수행 결과에 대한 무결성 등을 검증할 수 있다.

(3) 이동 에이전트는 에이전트 소유자 O에서 출발하여 호스트 H_1까지 수행하여 얻은 결과 값인 $((O, Request), S_0)$, $((H_1, O, Response_1), M_1, S_1)$을 가지고 다음 목적지인 호스트 H_2에 도착한다.

① 호스트 H_2는 이동 에이전트가 가지고 온 호스트 H_1의 서명 값과 O의 서명 값 둘 다에 대하여 검증을 수행한다. O의 서명 검증 방식은 (2)의 ①번 단계와 동일하며, 호스트 H_1의 서명 값 검증은 사용되는 서명 기법에 따라 적당한 방식으로 검증을 수행한다. 이 때, O의 서명 값을 생성할 때와 호스트의 서명 값을 생성할 때 서로 다른 서명 방식을 적용하는 이유는 O의 서명 값은 에이전트가 모든 작업 수행을 마치고 에이전트 소유자에게 되돌아갔을때 에이전트의 수행 결과를 평가하기 위해 필요하기 때문에 매 단계에서 동일한 서명 기법을 적용하여 계산되어야 하는 반면, 호스트의 서명 값은 단순히 호스트의 인증을 목적으로 하기 때문에 효율성이나 상황을 고려하여 적당한 서명 기법을 적용함으로써 프로토콜의 유연성을 높이고자 한다.

② $Hvalue_2 = hash(H_2, O, Response_2)$

검증한 O의 서명 값이 유효할 경우 호스트 H_2은 이동 에이전트를 실행하고 에이전트가 요청하는 *Request*에 대한 수행 결과인 $Response_2$을 생성한다. 서명 값을 생성하기 위해 자신의 ID인 H_2, 에이전트 소유자의 ID인 O, 호스트 H_2에서 생성한 $Response_2$을 암호학적 해쉬 함수를 적용하여 해쉬 값 $Hvalue_2$을 계산한다.

③ $M_2 = Sig_{pr_{H_2}}(Hvalue_2)$

개인키 pr_{H_2}을 적용하여 $Hvalue_2$에 대한 호스트 H_2의 서명 값을 생성한다.

④ $S_2 = S_1 \times S_0^{Hvalue_2} \bmod n_O$

호스트 H_2은 자신의 해쉬 값을 적용한 O의 서명 값을 생성한다.

(4) 이동 에이전트는 에이전트 소유자 O를 출발하여 $n-1$번째 호스트까지 수행하여 얻은 결과 값인 $((O, Request), S_0), ((H_1, O, Response_1), M_1, S_1), ((H_2, O, Response_2), M_2, S_2), \cdots, ((H_{n-1}, O, Response_{n-1}), M_{n-1}, S_{n-1})$를 가지고 호스트 H_n에 도착한다.

① 호스트 H_n은 에이전트가 가지고 온 값 중에서 이전 호스트의 서명 값을 검증하여 에이전트의 수행 결과에 대한 무결성을 검증할 수 있다.

② $Hvalue_n = hash(H_n, O, Response_n)$

서명 값을 검증한 후 이동 에이전트를 실행하고 에이전트의 요청에 대한 수행 결과 $Response_n$을 생성하고 자신의 ID인 H_n, 에이전트 소유자의 ID인 O, 호스트 H_n에서 생성한 $Response_n$을 암호학적 해쉬 함수를 적용하여 해쉬 값 $Hvalue_n$을 계산한다.

③ $M_n = Sig_{pr_{H_n}}(Hvalue_n)$

개인키 pr_{H_n}을 적용하여 $Hvalue_n$에 대한 호스트 H_n의 서명 값을 생성한다.

④ $S_n = S_{n-1} \times S_0^{Hvalue_n} \bmod n_O$

호스트 H_n은 자신의 해쉬 값을 적용한 O의 서명 값을 생성한다.

(5) 작업을 완료한 후 이동 에이전트는 O로 되돌아가며 이때 에이전트가 포함하는 수행 값은 $((O, Request), S_0), ((H_1, O, Response_1), M_1, S_1), \cdots, ((H_n, O, Response_n), M_n, S_n)$이다.

① 에이전트 소유자 O는 서명 값 $M_1, \cdots, M_n$에 대하여 올바른 서명인지 검증한다.

② $S_n = S_{n-1} \times S_0^{Hvalue_n} \bmod n_O$

호스트 H_1에서 H_n까지의 해쉬 값을 더한 O의 서명 값인 S_n에 대하여 자신의 개인 키를 적용하여 정당한 값인지 검증한다.

3.2 제안 프로토콜 분석

제안 프로토콜을 적용할 경우 전방향 무결성, 부인 방지, 이전 결과에 대한 정직한 호스트 및 에이전트 소유자의 부정 검출을 제공할 수 있다. <표 2>는 기존 방안과 제안 프로토콜을 비교한 것이다.

〈표 2〉 기존 방안과 제안 프로토콜의 비교

	특 징	안 전 성	효 율 성
[9] 방안	• 암호 프로토콜을 이용한 부정 검출 • 실행 단계별 추적 불가	• 전자서명 및 해쉬 함수의 안전성에 기반 • 무결성, 부인 방지 제공	• 호스트 수 및 실행 단계에 비례하여 효율성 감소 • 추적 정보 저장을 위한 별도의 공간 필요
[10] 방안	• 부분 수행 결과에 대한 캡슐화	• 암호 알고리즘의 안전성에 기반 • 비밀성 제공, 알려진 평문 공격에 강함	• 암호화에 시간이 많이 걸림 • 적은 저장 공간 필요
[11] 방안	• 해쉬 체인을 이용한 수행 결과 보호	• 해쉬 함수의 안전성에 기반 • 비밀성, 무결성 제공	• 호스트 수에 비례하여 효율성 감소
제안 프로토콜	• 해쉬 체인을 적용한 부정 검출 및 수행 경로가 보호 • 실행 단계별 추적 가능	• 전자서명 및 해쉬 함수의 안전성에 기반 • 무결성, 부인 방지, 전방향 무결성 제공	• 호스트 수 및 실행 단계에 비례하여 효율성 감소 • 추적 정보 저장을 위한 별도의 공간 필요

다음은 전방향 무결성[13]에 대한 정의이다.

Definition 1. 전방향 무결성(Forward Integrity)은 이동 에이전트가 n개의 호스트 $H_1, H_2, \cdots, H_n$을 방문하고, 최초로 나타나는 악의적인 호스트가 H_c라 할 때, 이동 에이전트가 현재 방문하고 있는 호스트 H_i와 이전에 방문하였던 호스트에서 생성된 결과 값이 변경될 수 없다는 것을 의미한다(단, $i < c$).

제안 프로토콜의 각 호스트에서 계산되는 값은 전자 서명되기 때문에 서명 방안이 안전할 경우, 악의적인 호스트는 이전에 생성된 결과 값들에 대하여 수정 공격이 불가능하므로 전방향 무결성이 제공되며, 같은 이유로 부인 방지가 제공된다.

또한 에이전트를 수행하는 호스트는 이전 호스트에서 수행된 이동 에이전트의 수행 결과에 대해서 서명 값 검증을 통해 부정 검출을 하는 것이 가능하다.

끝으로, 에이전트 소유자는 이동 에이전트가 저장하고 있는 이전 호스트들의 RSA 서명 값을 검증하고 자신의 서명 S_n값을 계산함으로써 이전 결과에 대한 부정을 검출할 수 있다. 이때 $S_0 = Hvalue_0^{pr_O} \bmod n_O$이고 S_i의 계산 식은 다음과 같다.

$$S_i = \begin{cases} S_0^{Hvalue_1} \bmod n_O & i = 1 \\ S_{i-1} \times S_0^{Hvalue_i} \bmod n_O & i > 1 \end{cases}$$

O의 개인키 pr_O가 안전하다면 S_n값은 최종적으로 O만 계산할 수 있다. S_n값의 계산 과정은 <표 3>과 같다.

〈표 3〉 S_n값

$$S_n = S_{n-1} \times S_0^{Hvalue_n} \bmod n_O \quad \cdots (1)$$
$$= S_{n-2} \times S_0^{Hvalue_{n-1}} \times S_0^{Hvalue_n} \bmod n_O \quad \cdots (2)$$
$$= S_0^{Hvalue_1} \times \cdots \times S_0^{Hvalue_{n-1}} \times S_0^{Hvalue_n} \bmod n_O \quad \cdots (3)$$
$$= (S_0)^{Hvalue_1 + \cdots + Hvalue_n} \bmod n_O \quad \cdots (4)$$
$$= (Hvalue_0^{pr_O})^{Hvalue_1 + \cdots + Hvalue_n} \bmod n_O \quad \cdots (5)$$
$$= (Hvalue_0^{Hvalue_1 + \cdots + Hvalue_n})^{pr_O} \bmod n_O \quad \cdots (6)$$

S_n값은 (1)의 식으로 정의된다. 이 값은 (2)에서 보는 바와 같이 에이전트 소유자 O에서의 S_0 값에 대한 해쉬 값 $Hvalue_0$, 에이전트가 이동한 첫 번째 호스트의 S_1 값에 대한 해쉬 값 $Hvalue_1, \cdots$, 에이전트가 이동한 마지막 호스트의 S_n 값에 대한 해쉬 값 $Hvalue_n$ 각각의 합으로 바꿔 적을 수 있다. 각 단계의 계산 과정을 통해 S_n 값은 최종적으로 (6)의 식으로 변형되고 모든 해쉬 값의 합에 대해 에이전트 소유자 O의 개인키로 서명된 형태가 된다.

4. 결론 및 향후 연구

본 논문에서는 클라이언트-서버 모델의 취약점을 개선하기 위한 방안 중 하나로 제안된 이동 에이전트 시스템의 알려진 위협을 분석하고 악의적인 호스트로부터 이동 에이전트의 수행 결과에 대한 침해가 있었는지의 여부를 판단하기 위한 이동 에이전트 수행 결과 보호 프로토콜을 제안하였다. 이 프로토콜에서 이동 에이전트는 RSA 전자 서명을 기반으로 수행 과정에서 얻는 모든 해쉬 값을 더함으로써 일종의 해쉬 체인(Hash Chaining)을 형성하고 있다. 따라서 RSA 전자 서명 값과 모든 해쉬 값을 더한 값을 검증함으로써 정당한 호스트와 에이전트 소유자는 에이전트의 수행 결과에 대한 변경 유무를 판단할 수 있으며 정당한 호스트라 하더라도 자신이 한번 생성하여 다음 호스트에게 넘겨준 해쉬 값과 서명 값에 대해서 변경 또는 삭제를 수행 할 수 없도록 되어 있다.

그러나, 제안 프로토콜은 이동 에이전트가 포함하는 데이터를 누구나 읽을 수 있다는 특징을 가지고 있기 때문에 보다 강력한 보호를 필요로 하는 분야에서 사용하기 위해서는 데이터의 비밀성(Privacy)이 제공되어야 한다. 따라서, 이 부분에 대한 추가적인 연구를 진행할 계획이다.

현재 많은 이동 에이전트 시스템이 개발되고 상용화되어 있는데, 보다 폭넓은 응용 분야 진출을 위해서는 시스템을 안전하게 구현할 필요가 있다. 에이전트 시스템을 보호하기 위한 방안은 기존의 호스트 또는 네트워크 상에서의 공격에 대한 보호 방안을 적용할 경우 대부분 해결이 가능한 반면 이동 에이전트 자체를 보호하기 위한 방안에 대한 연구는 아직 미흡한 실정이다. 시스템의 전체 보안 강도를 결정짓는 것은 시스템의 가장 약한 부분이기 때문에 이동 에이전트를 보호하기 위한 방안에 대한 연구는 꾸준히 계속 되어야 한다.

참 고 문 헌

[1] John E. Canavan, "Fundamentals of Network Security," Artech House. INC, 2001.

[2] 김영대, "자바로 구현하는 이동 에이전트 시스템", 프로그램 세계, 2000.

[3] C. Ghezzi and G. Vigna, "Mobile Code Paradigms and Technologies : A Case Study," in Proceeding of the First International Workshop on Mobile Agents (MA '97), 1997.

[4] Jim Waldo, "Mobile code, distributed computing, and agents," IEEE Intelligent Systems, 2001.

[5] Wayne A. Jansen, "Countermeasures for Mobile Agent Security," Elsevier Science, 2000.

[6] 신정화, "이동코드 보안 모델에 관한 연구", 부경대학교 대학원 석사학위논문, 2000.

[7] 신 원, "안전한 이동 에이전트 시스템의 설계와 응용", 부경대학교 대학원 박사학위논문, 2001.

[8] Michael S. Greenverg, jennifer C. Byington, "Mobile Agents and Security," IEEE Communications Magazine, 1998.

[9] Panayiotis Kotzanikolaou, Mike Burmester, Vassilios Chrissikopoulos "Secure Transactions with Mobile Agents in Hostile Environments," ACISP 2000, pp.289-297.

[10] A. Young, M. Yung, "Sliding Encryption : A Cryptographic Tool for Mobile Agents," Proc. 4th Int'l Wksp, Fast Software Encryption, FSE.

[11] G. Karjoth, N. Asokan and C. Gulcu, "Protecting the Computation Results of Free-roaming Agents," Proceedings of the Second international Workshop, MA'98. pp.195-207, 1998.

[12] Alfres J. Menezes, Paul C. vaz Oorschot, Scoot A. Vanstone, "Handbook of Applied Cryptography," CRC.
[13] Bennet S. Yee, "A Sanctuary for Mobile Agents," Secure Internet Programming, LNCS 1603, Springer-Verlag, pp. 261-274, 1999.
[14] G. Vigna, "Cryptographic Traces for Mobile Agents," In : G. Vigna (Ed.). Mobile Agents and Security, Springer-Verlag, Lecture Notes in Computer Science 1419, pp.137-153, 1998.
[15] 신 원, 이경현, "이동 에이전트 실행 보호를 위한 암호학적 추적 방안", 통신정보보호학회논문지, 제11권 제3호, 2001.

김 희 연

e-mail : anilo97@korea.com
2001년 부경대학교 전자계산학과 졸업 (이학사)
2001년~현재 부경대학교 전자계산학과 대학원 석사 과정
관심분야 : 정보보호, 이동에이전트, 암호이론, 공개키 기반구조

신 정 화

e-mail : shinjh@unicorn.pknu.ac.kr
1997년 한국방송통신대학교 전자계산학과 졸업(이학사)
2000년 부경대학교 전산정보학과 대학원 졸업(이학석사)
2002년~현재 부경대학교 전자계산학과 대학원 박사과정
관심분야 : 정보보호, 이동에이전트, 암호이론, 네트워크 보안

신 원

e-mail : shinweon@ahnlab.com
1996년 부산수산대학교(현 부경대학교) 전자계산학과 졸업(이학사)
1998년 부경대학교 전자계산학과 대학원 졸업(이학석사)
2001년 부경대학교 전자계산학과 대학원 졸업(이학박사)
2002년~현재 안철수 연구소
관심분야 : 정보보호, 이동에이전트, 멀티미디어 통신, 암호이론, 네트워크 보안

이 경 현

e-mail : khrhee@pknu.ac.kr
1982년 경북대학교 사범대학 수학교육과 졸업(이학사)
1985년 한국과학기술원 응용수학과 졸업 (이학석사)
1992년 한국과학기술원 수학과 졸업(이학박사)
1985년~1993년 한국 전자 통신 연구소 연구원, 선임 연구원
1993년~현재 부경대학교(구 부산수산대학교) 전임강사, 조교수, 부교수
1995년~1996년 Univ. of Adelaide, 응용수학과, Australia 방문교수
1999년 Univ. of Tokyo, 객원 연구원
1997년~현재 한국멀티미디어학회 학술이사
2001년~현재 한국통신정보보호학회 논문지 편집위원
관심분야 : 정보보호론, 멀티미디어 정보보호, 네트워크성능 평가, 암호학, 재시도 대기체계론