익명성을 보장하는 이동 에이전트기반 상호 인증 구조

o
백광진*, 서래원**
고려대학교 컴퓨터학과*
배재대학교 컴퓨터공학과**

Anonymity Warranting Mobile Agent Systems: A Mutual Authentication Scheme for Agent & Server

o
Kwang-Jin Paek*, Seo Laiwon**
Dept. of Computer Science & Engineering, Korea University*
Paichai University, Department of Computer Engineering**

요 약

이동 에이전트 시스템은 인터넷에서 증가하고 있으며 전자 상거래 분야에 있어서 기반이 되는 개념이다. 전자 상거래뿐만 아니라 다른 많은 분야에서 보안은 중요한 문제이다. 기존의 보안 기법을 이동 에이전트 시스템에 적용하기에는 많은 어려움이 있으므로 이동 에이전트 시스템에 적합한 새로운 보안 프로토콜을 필요로 하고 이에 대한 많은 연구가 진행 중에 있다. 기존 시스템과 프로토콜은 서버나 클라이언트 중 한 쪽만 보호하는 단방향 인증을 제공하고 에이전트의 중요 정보를 공개하고 수명(liveness)을 보장하지 못하는 한계점을 지니고 있다. 본 논문에서 제시하는 시스템과 인증 프로토콜은 기존의 한계점을 해결하기 위한 방안을 제시하고 에이전트 서버간 상호 익명성을 보장하며 서버의 결함을 허용하는 융통성도 제공한다. 상호 인증 프로토콜을 제공함으로서 불법적인 에이전트와 서버의 침입을 막을 수 있다.

1. 서 론

이동 에이전트를 지원하는 시스템은 인터넷에서 점차 증가하고 있으며 이동 에이전트 시스템은 전자 서비스를 위한 기본 요소로서 인식되며 특히, 전자 상거래 영역에서 그 필요성이 부각되고 있다. 이러한 응용 프로그램 영역에서 보안의 중요성은 모든 요소보다 우선되어야한다.

일반적으로 이동 에이전트 시스템에서 보안은 두 관점에서 볼 수 있다. 이동 에이전트와 서버의 입장에서 서로를 신뢰하기 위한 여러 가지 연구가 진행되고 있다. 대부분의 연구는 불법적인 호스트로부터 합법적인 에이전트를 보호하거나 또는 그 반대의 경우를 가정한 연구가 있어왔다. 일방적으로 한 쪽만을 보호하기 위한 제안들은 많이 찾아 볼 수 있다. 그러나 불법적인 호스트로부터 이동 에이전트를 보호하는 것은 불법적인 에이전트로부터 호스트를 보호하는 것만큼이나 중요하다.[3]

안전(safety) 문제보다 더 어려운 문제는 에이전트가 계획된 여정을 통해 임무를 완수할 수 있도록 수명(liveness)을 보장하는 것이다. 새로 제시될 프로토콜은 서버가 멈추거나 네트워크가 단절되더라도 에이전트의 수명을 보장할 수 있어야 한다. 바이러스와 같은 이동 에이전트는 모든 서버를 파괴시킬 수 있다. 새로 제시될 프로토콜은 불법적인 에이전트를 막기 위한 이동 에이전트의 인증도 지원해야한다.

본 논문에서는 이동 에이전트에서 상호 보안을 제공하고 수명을 보장하는 프로토콜을 제시할 것이다.

본 논문은 다음과 같이 구성된다. 2장에서는 기존의 이동 에이전트 시스템의 보안을 위해 제안된 프로토콜과 시스템을 특징에 따라 분류하여 살펴보고 3장에서는 본 논문에서 제시하는 프로토콜에서 사용될 암호화 기술을 살펴보고 4장에서는 본 논문에서 제시된 프로토콜과 기존의 다른 프로토콜의 장단점을 평가한다. 마지막으로 5장에서 결론을 내리고 향후 연구 과제를 제시한다.

2. 이동 에이전트 시스템을 위한 보안 프로토콜

본 장에서는 이동 에이전트의 보안 문제에 있어서 기본이 되는 개념과 한계점을 살펴보고 기존 시스템을 이동 에이전트와 서버 입장에서 각각 나누어 살펴볼 것이다.

이동 에이전트 보안 문제에 있어서 한계점[4]은 다음과 같다.

- 암호화 기법을 사용하지 않고 에이전트가 포함하고 있는 정보를 보호할 수 없다.

-신뢰성을 제공하는 하드웨어 없이 에이전트의 변형을 막을 수 없다.

-새로 이주해온 에이전트가 바이러스가 아니라는 완벽한 보장을 할 수 없다.[5]

-에이전트의 정책이 논리적이고 정확하더라도 여러 객체와의 복잡한 상호작용의 결과가 정확할 것이라는 완벽한 보장을 할 수 없으며[6], 이 분야는 계속적인 연구가 진행중이다.

2.1. 이동 에이전트를 위한 보안 프로토콜

Fritz Hohl의 논문[1]에서는 불법적인 서버로부터 에이전트를 보호하기 위해 다음과 같은 두 가지 메커니즘을 제시한다. 1)Code Mess-Up과 2)Limited Lifetime of Code and Data이다. 기본 개념은 침입자에게 코드를 분석할 충분한 시간을 주지 않는 것이다. Code Mess-Up 기법은 비암호학적인 방식으로 분석하는데 시간이 걸리도록 코드의 구조를 바꾸는 것이다. Limited Lifetime of Code and Data 기법은 데이터와 발행인 및 유효시간에 전자서명을 하는 것이다. 이렇게 함으로써 손상된 에이전트의 계속적인 사용을 막을 수 있다.

Wilhelm, Uwe, G.과 Staamann, Sebastian의 논문[2]은 에이전트를 보호하기 위한 두 가지 요소를 제공한다. 1)TPE(tamper-proof environment)과 2)CryPO(cryptographically protected objects) 프로토콜이 사용된다. TPE는 완전한 컴퓨터로서 에이전트가 실행되기 위한 환경을 제공하는 가상 머신(VM: virtual machine)을 실행한다. TPE는 자신만이 알고 있는 개인키를 가지고 있다. TPE의 소유자도 TPE의 개인키는 알 수 없다. TPE가 지닌 정보는 직접적인 접근이 불가능하다. CryPO 프로토콜은 실행 가능한 에이전트를 암호화하여 TPE로 전송하여 에이전트가 안전하게 수행되도록 한다.

<그림 1>은 초기화 과정으로 TM은 자신의 인증된 개인키를 발행하고 TPE의 공개키에 사인을 해서 AE에게 보낸다. AE는 다른 중개자(broker)에게 자신의 참조(reference)를 등록한다.

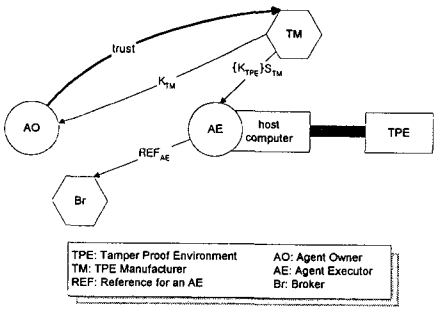

그림 1. CryPO 프로토콜의 초기화[2]

<그림 2>는 CryPO 프로토콜의 활용 과정을 보여준다. 초기화 과정을 마친 후에 AO가 에이전트를 TPE의 공개키로 암호화하여 AE에게 전송함으로써 에이전트 시스템을 사용할 수 있다.

James Riordan 과 Bruce Schneier의 논문[7]에서는 서버의 시간과 환경에 의해 키를 생성해 암호화된 에이전트를 실행할 수 있도록 한다. 각 서버는 미리 환경키를 생성해서 에이전트를 암호화할 때 사용할 수 있도록 한다.

2.2. 서버를 위한 보안 프로토콜

Ordille, Joann의 논문[3]에서는 에이전트에 대한 서버의 위협은 무시하고 불법적인 에이전트로부터 서버를 보호하기 위한 방안을 제시한다. 첫째는 각각의 새로운 서버 S_i는 에이전트의 경로 목록에 자신을 추가시킨다. 그리고 다음 서버인 S_{i-1}에게 자신이 전자 서명한 경로 목록을 제공한다. 다음 서버는 경로에 있는 모든 서버를 신뢰할지 그리고 에이전트가 올바르게 전달되고 인접한 이전의 서버가 믿을 만한지를 결정하게 된다. 일반적으로 서버는 이전의 모든 서버에 대한 신뢰 검증을 원하지 않는다. 두 번째 방법은 서버 S_i가 에이전트를 다른 서버 S_{i-1}으로 보낼 때 한 홉(hop)단위로 전자 서명을 한다. 한 홉 $S_i \rightarrow S_{i-1}$에 전자 서명을 하여 전송한다. 첫 번째 방식은 경로가 증가할 수 록 검증에 과중한 부담이 들지만 두 번째 방식은 홉단위로 검증을 할 수 있으므로 부담을 덜 수 있다.

2.3. 기존 시스템의 분석

본 장에서 살펴본 에이전트 시스템의 단점을 분석해보면 다음과 같이 요약할 수 있다.

- 에이전트와 서버간 상호 인증을 제공하지 않는다.
- 에이전트의 비밀 정보가 노출된다.
- 익명성을 제공하지 못한다.
- 시스템 결함을 허용하지 않는다. 즉, 이동 에이전트의 수명을 보장하지 못한다.

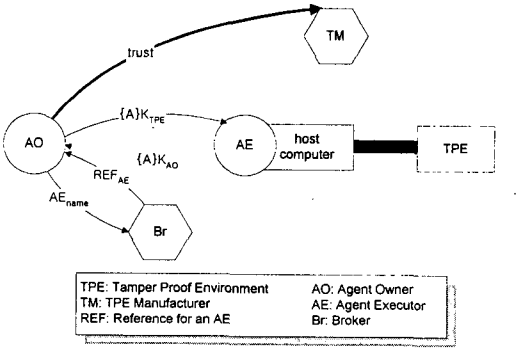

그림 2. CryPO 프로토콜의 활용[2]

3. 서버와 에이전트간 상호 인증 구조

본 논문에서 제시하는 익명성을 제공하는 상호 인증 시스템의 구성은 <그림 3>과 같다.

다음은 에이전트의 이동 과정을 보여준다.

1. AO→ABS: $\{mc\}CK_{AO,ABS}$
2. ABS→AP1: $\{mc'\}CK_{ABS,AP1}$
3. AP1→ABS: $\{mc'^{AP1}\}CK_{ABS,AP1}$
4. ABS→AP2: $\{mc'\}CK_{ABS,AP2}$

다음은 AO와 ABS간의 키 분배 과정을 보여준다.

a. AO→ABS: $\{\{A, T\}SK_{AO}\}PK_{ABS}$
b. ABS→AO: $\{\{CK_{AO,BS}, ABS, T\}SK_{ABS}\}PK_{AO}$

다음은 ASB와 AP1간의 키 분배 과정을 보여준다.

c. ABS→AP1: {{ABS, T}SK_{ABS}}PK_{AP1}

d. AP1→ABS: {{$CK_{BS,AP1}$, AP1, T}SK_{AP1}}PK_{ABS}

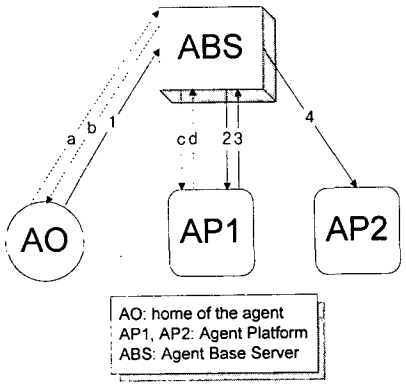

그림 3. 이동 에이전트 경로 및 키 분배 과정

T는 타임스탬프(timestamps)를 나타내며 타임스탬프를 사용함으로써 통신키(CK)의 재사용을 막을 수 있을 뿐만 아니라 two-step handshake를 생략할 수 있다.[8]

제안된 시스템은 에이전트 생성자(AO: Agent Owner)와 에이전트 베이스 서버(ABS: Agent Base Server) 그리고 에이전트 플랫폼(AP: Agent Platform)으로 구성된다.

AO는 에이전트를 실행하기 위해 ABS에게 의뢰하여 에이전트를 연결키인 $CK_{AO,ABS}$로 암호화하여 AP에게 전송한다. AO는 에이전트의 모든 실행권한을 ABS에게 넘겨준다. AO는 어느 AP에서 에이전트가 실행될지 알 수 없다.

ABS는 AO로부터 받은 에이전트를 무사히 실행할 책임을 지게된다. ABS는 AP에게 에이전트의 중요한 데이터는 넘겨주지 않고 실행코드와 그에 필요한 데이터만을 넘겨준다. ABS는 AP 리스트를 관리하고 에이전트의 수명을 보장하기 위해 계속적인 감시를 한다. ABS는 에이전트의 핵심 정보를 관리하며 AP가 에이전트를 손상시켜도 복구시킬 수 있으며 다른 AP으로의 전송을 책임진다. 즉, 에이전트가 임무를 끝낼 수 있도록 한다.

AP는 에이전트의 이동 경로를 알 수 없으며 AO를 식별할 수도 없으며 단지 ABS에게 실행결과를 전송해야하는 것만을 알고 있다.

4. 기능 비교 평가

본 장에서는 본 논문에서 제시한 시스템과 기존 시스템을 여러 면에서 비교해 성능을 평가한다.

제시된 시스템과 프로토콜은 기존 시스템에서 간과한 수명을 보장하기 위해 ABS를 두어서 다른 AP로 이동하려만 반드시 ABS를 통해 간접적으로 이동을 해야만한다. 이 구조는 AO와 AP간의 익명성을 제공하며 ABS가 에이전트의 중요 부분을 관리함으로써 AP의 결함을 허용하는 장점도 제공한다.

제시된 프로토콜은 상호 인증을 제공함으로써 불법적인 에이전트와 서버의 침입을 방지할 수 있다.

표 1. 시스템 및 프로토콜의 기능 비교

항 목	[1]	[2]	[7]	제안된 시스템
보호 방식	시간 제한 코드 혼합	하드웨어	환경키	정보 은닉
통신키 암호화 방식	비대칭	비대칭	비대칭	비대칭
에이전트 이동 방식	1 pass 직접	1 pass 직접	1 pass 직접	2 pass 간접
경로 공개	○	○	○	×
결함 허용	×	×	×	○
익명성 지원	×	×	×	○
인증 방식	단방향	단방향	단방향	양방향

5. 결 론

본 논문에서 에이전트 시스템을 위한 보안 프로토콜을 살펴보고 그 한계를 극복하기 위한 개선된 시스템과 프로토콜을 제시하였다. 기존 시스템과 프로토콜에 비교해서 익명성과 상호 인증 그리고 수명 보장의 장점을 제공한다. 향후 과제로서 본 논문에서 제시된 시스템을 다른 시스템과 성능 평가를 하는 것이며 ABS에 집중된 부담을 분산시키고 이동 에이전트의 보다 자유로운 이동을 지원하기 위한 연구가 필요하다.

참고문헌

[1] Hohl, Fritz, "An approach to solve the problem of malicious hosts," Universitat Stuttart, Fakultat Informatik, Fakultatsbericht Nr.1997/03.

[2] Wilhelm, Uwe, G. and Staamann, Sebastian, "Protecting the Itinerary of Mobile Agents," in Proceedings of the ECOOP Workshop on Distributed Object Security and 4th Workshop on Mobile Object Systems: Secure Internet Mobile Computations, pp. 135-145, INRIA, France, 1998.

[3] Ordille, Joann, "When agents roam, who can you trust?" In Proc. of the First Conference on Emerging Technologies and Applications in Communications, Porland, May 1996.

[4] David Chess, Benjamin Grosof, Colin Harrison, David Levine and Colin Parris, "Itinerant Agents for Mobile Computing," IBM Research Report RC20010, IBM, March 1995.

[5] F. Cohen, "Computer viruses: Theory and experiment," Computers & Security, June 1987.

[6] J. Kephart, T. Hogg, and B. A. Huberman, Can predictive agents prevent chaos? In Economics and Cognitive Science, Pergamon Press, Oxford, 1991.

[7] James Riordan and Bruce Schneier, "Environmental Key Generation towards Clueless Agents," in Giovanni Bigna(Ed.), Mobile Agents and Secrity, pp15-24, Springer-Verlag, 1998.

[8] D. E. Denning and G. M. Sacco, "Timestamps in key distribution protocols," Commun. ACM, vol. 24, pp.533-535.