이동 에이전트 보안을 위한 공유 트러스티드 플랫폼 설계

송상훈[0] 노용덕
세종대학교 컴퓨터공학과
(song[0], novak)@sejong.ac.kr

Design of a Shared Trusted Platform for Mobile Agents Security

Sanghoon Song[0] Youngdeok Noh
Dept. of Computer Engineering, Sejong University

요 약

이동 에이전트는 네트워크 상에서 필요에 따라 자발적으로 이종의 호스트들을 이동하면서 정해진 작업을 실행하고 결과를 제공하는 프로그램인데, 온라인 쇼핑, 실시간 장치제어, 분산처리 작업 등에 응용할 수 있는 기술이다. 그러나 이동 에이전트의 보안 문제등 해결해야 될 여러 가지 문제 점들을 안고 있다. 본 논문은 이동 에이전트 보안을 위한 트러스티드 플랫폼 비용, 설치 및 운영의 어려운 점을 해결하기 위하여 트러스티드 플랫폼을 공유하여 서비스 제공자들이 쉽게 이동 에이전트를 위한 신뢰할 수 있는 플랫폼 서비스를 제공할 수 있도록 하는 방법을 제안한다.

1. 서 론

인터넷의 발달로 인해 데이터뿐 만 아니고 디바이스도 이동성을 갖게 되었고, 자바 프로그램 언어는 소프트웨어도 이동성을 갖을 수 있다는 것을 보여준다. 컴퓨터 네트워크 상에서 한 노드에서 다른 노드로 자발적으로 이동하면서 사용자를 대신하여 작업을 실행하는 프로그램을 이동 에이전트라고 하는데, 온라인 쇼핑, 실시간 장치 제어, 분산처리 작업 등 여러 가지 분야에 응용할 수 있는 기술이다.

미리 정해진 작업을 위해 이동 에이전트는, 미리 정해진 경로 또는 이동 에이전트에 의해 입수된 정보에 따라 동적으로 결정되는 경로를 따라 노드들을 방문하여 작업을 수행해 나간다. 작업 수행이 완료 되면 홈 사이트로 돌아와서 결과를 보고하거나 또는 홈 사이트에 보고할 필요가 없는 경우는 자동으로 이동 에이전트를 없어지게 하여 끝난다.

네트워크 트래픽 및 지연을 감소시키고 연결 상태가 좋지않은 이동 장치에 효율적인 서비스를 제공할 수 있는 등 다양한 장점을 제공할 수 있다. 그러나 이동 에이전트의 큰 문제점의 하나로서 호스트와 이동 에이전트 간의 보안 문제가 이동 에이전트 응용의 확산을 막고있다. 먼저 호스트 측의 보안 문제로서 이동되어 들어온 에이전트가 바이러스 등과 같이 유해한 코드가 아니라는 것을 확인하는 것이 쉽지가 않다는 것이다. 효율적인 접근제어에 의한 고전적인 호스트 보안 기술들이 제안되어 왔다. 이에 반하여 이동 에이전트 측의 보안은 유해 호스트로부터 이동 에이전트를 보호하는 문제이다. 호스트는 다른 호스트로부터 이주해온 한 이동 에이전트에 대한 모든 제어가 가능하기 때문에 유해 호스트에 들어간 이동 에이전트를 보호하는 것은 현실적으로 해결책이 없는 것으로 알려져 있다. 이동 에이전트와 플랫폼의 특성상 아직까지 적용하기에 어려운 몇 가지 이론적인 보안 기술과 트러스티드 플랫폼을 설계하는 것 외에는 별다른 방법이 제안되지 않고 있다[1].

본 논문은 이동 에이전트 보안을 위한 트러스티드 플랫폼 비용, 설치 및 운영의 어려운 점을 해결하기 위하여 트러스티드 플랫폼을 공유하여 서비스 제공자들이 쉽게 이동 에이전트를 위한 플랫폼을 제공할 수 있도록 하는 방법을 제안하는 것이다.

2. 트러스티드 플랫폼

트러스티드 플랫폼은 CA(Certification Authority)와 같이 제 3의 신뢰할 수 있는 기관에서 설치하여 믿을 수 있고, 시스템이 침입자에 의해 쉽게 변조될 수 없게 설계되어 이동 에이전트들의 실행을 보장할 수 있는 플랫폼이다 [2][3].

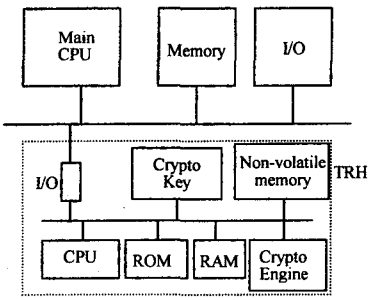

그림 1. 트러스티드 플랫폼

그림 1 과 같이 트러스티드 플랫폼은 일반적인 컴퓨터

요소 외에 TRH(Tamper Resistant Hardware) 부분을 추가하여 구성된다. TRH는 플랫폼 소프트웨어들에 대한 해쉬 값들을 저장하고 있는데, 부팅 시 해쉬 값을 사용하여 플랫폼 소프트웨어의 무결성(integrity) 검사를 하게 된다. 새로운 시스템 프로그램 또는 응용프로그램을 설치할 때마다 관리자에 대한 인증 절차 후에 새로운 프로그램에 대한 해쉬를 만들고 TRH내에 저장한다. 시스템의 무결성 체크를 원할 때마다 TRH의 기능을 이용하여 수행할 수 있다. 즉, 데이터가 적재될 때, 또는 응용 프로그램을 실행하기 위해 적재할 때 무결성 검사를 할 수 있다.

이동 에이전트가 다음 플랫 폼으로 이동 시에 우선 목적지 플랫폼의 인증과정을 거쳐서 트러스티드 플랫폼이 확인 되면 이동시킨다. 이동 시에 데이터를 자신의 비밀키로 암호화하고 이를 에이전트 코드에 첨부하여 목적지 플랫폼의 공개키로 암호화하여 보낸다. 목적지 플랫폼의 TRH 는 수신된 패킷의 암호문을 풀어서 플랫폼에 에이전트 코드를 전달하고 민감한 데이터는 TRH 내에 저장한다. 물리적 공격에 의해 TRH가 공개될 때에 TRH 자신의 개인키는 자동적으로 지워지도록 설계되고 민감한 데이터들은 자료 손실을 방지하기 위하여 내부의 비휘발성 메모리에 저장된다. 이들은 TRH 자신의 개인 키로 암호화하여 저장하기 때문에 권한을 부여받은 관리자에 의해 TRH의 개인 키를 이용하여 복구할 수 있다.

3. 트러스티드 플랫폼 공유

이동 에이전트를 이용하는 응용 프로그램들이 활성화되기 위해서는 이동 에이전트가 서버 측을 신뢰할 수가 있어야 한다. 신뢰되지 않는 서버에서 이동 에이전트의 트랜잭션 결과를 사용자가 믿을 수가 없으므로, 서비스를 제공하는 서버의 신뢰는 사용자에게 중요하다. 아직까지 이동 에이전트 보안에 대한 해결책은 트러스티드 플랫폼을 각 서버에 설치하는 것 외에 좋은 방법이 알려져 있지 않다.

앞 절에서 설명된 트러스티드 플랫폼을 트러스티드 플랫폼 설치 기관(Certified Authority)에서 이동 에이전트에 서비스를 제공하는 모든 이종의 일반 서버들에 설치를 하는 것은 쉽지 않을 뿐만 아니라, 각 서비스 제공자들에게도 비용 부담이 된다. 서비스 제공 서버 수가 많아지면 설치기관에서 이들 서버에 하드웨어 설치 및 인증서 발급 및 취소 등의 관리 작업이 어려워진다. 그러나 일부 ISP 등과 같이 신뢰할 수 있는 소수의 제 3 자 사업자에게만 설치하는 것은 비용, 그리고 관리하는 면에서 장점이 있다.

일반적으로 거래 금액이 큰 시장 거래는 사기 등의 위험을 방지하기 위해 보통 중개인을 통하여 하듯이, 이동 에이전트도 브로커 역할을 할 수 있는 제삼자의 트러스티드 플랫폼에서 트랜잭션을 이루어지게 하는 것이 이동 에이전트 사용자에게 더욱 신뢰를 줄 수 있다. 따라서 트러스티드 플랫폼을 모든 서버에 설치하지 않고, 신뢰할 수 있는 제3의 서버에 설치를 하고 각 서비스 제공자는 서비스 에이전트를 트러스티드 플랫폼에 정착시키거나 필요에 따라 이동시켜 신뢰할 수 있는 서비스를 제공할 수 있다.

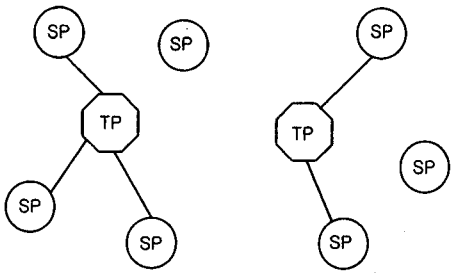

그림 2. 공유 트러스티드 플랫폼 구조

그림 2는 공유 트러스티드 플랫폼(Shared TP)과 서비스 제공자(Service Provider)의 구조를 보여준다. 서비스 제공자와 공유 트러스티드 플랫폼사이의 실선은 서비스 제공자가 트러스티드 플랫폼에 등록된 것을 나타내는데, 이 정보는 디렉토리 서비스에서 제공한다. 하나의 서비스 제공자는 다수의 트러스티드 플랫폼에 등록하여, 사용자 에이전트가 이동거리가 짧은 플랫폼에서 서비스를 받을 수 있도록 할 수 있다. 트러스티드 플랫폼과 연결이 안된 서비스 제공자는 보안이 요구되지 않는 서비스를 제공 하는 경우이다.

사용자의 이동 에이전트는 서비스 에이전트가 있는 트러스티드 플랫폼으로 이동하여 해당 서비스 에이전트와 트랜잭션을 처리하고 다른 서비스 제공자의 서비스를 받기 위해 해당 공유 트러스티드 플랫폼으로 이동하게 된다. 이동 에이전트가 원하는 서비스 에이전트의 트러스티드 플랫폼 정보는 디렉토리 서비스를 통하여 얻을 수 있다.

인증기관은 해당 트러스티드 플랫폼에 대한 비밀키와 공개키를 만들어 공개키에 대한 인증서와 함께 디렉토리 서비스에 등록을 한다. 이 공개키는 이동 에이전트가 이동 중에 비밀성을 유지하기 위해 암호화해야 하는데 이를 위한 세션 키를 만드는데 사용되거나, 트러스티드 플랫폼을 인증하는데도 사용된다.

그림 3은 서비스 제공자(SP)가 원하는 공유 트러스티드 플랫폼(STP)을 선택하여 서비스 에이전트의 트러스티드 플랫폼으로 등록하는 단계들을 보여주고 있다. 서비스 제공자는 한 개 이상의 트러스티드 플랫폼에 등록이 가능하여 이동 에이전트는 이동거리가 짧은 트러스티드 플랫폼을 선택할 수 있다.

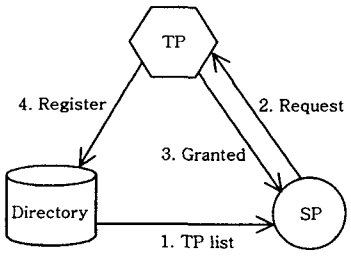

그림 3. 공유 트러스티드 플랫폼 등록

등록 단계는 다음과 같이 이루어진다.

단계 1. 서비스 제공자는 디렉토리 서비스를 통하여 트러스티드 플랫폼의 리스트를 얻는다.
단계 2. 서비스 제공자는 트러스티드 플랫폼에 계정을 요청한다.
단계 3. 트러스티드 플랫폼은 서비스 제공자에 대한 요청을 수락하고 서비스 제공자의 계정을 위한 작업을 한다.
단계 4. 트러스티드 플랫폼은 디렉토리 서비스에 서비스 제공자의 트러스티드 플랫폼정보의 등록을 요청한다. 서비스 제공자는 이용가능한 트러스티드 플랫폼들의 리스트를 디렉토리 서비스를 통하여 얻게 된다.

이동 에이전트가 서비스 제공자와의 트랜잭션에서 안전한 서비스를 원할 때, 해당 서비스 제공자가 안전한 서비스를 제공할 수 있는 트러스티드 플랫폼으로 이동하여 서비스를 받게된다. 다음 그림 4는 이동 에이전트의 이동 단계들을 보여준다.

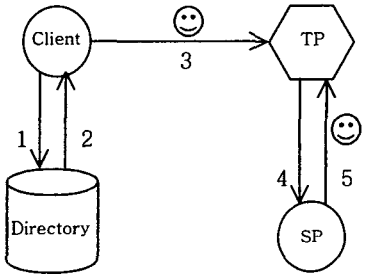

그림 4 이동 단계

단계 1. 이동 에이전트는 디렉토리 서비스를 통하여 서비스 제공자의 트러스티드 플랫폼에 관한 정보를 요청한다.
단계 2. 디렉토리 서비스는 서비스 제공자에 대한 트러스티드 플랫폼에 관한 정보를 제공한다.
단계 3. 이동 에이전트는 트러스티드 플랫폼으로 이동한다. 클라이언트 인증을 위해 이동 코드에 대한 암호학적 해쉬에 대한 디지털 서명을 첨부한다. 물론 이동 중에 보안을 위하여 트러스트디 플랫폼의 공개키로 암호화한다. 트러스티드 플랫폼은 암호화된 이동 에이전트를 복호화하고, 디지털 서명을 통해 클라이언트에 대한 인증을 한다.
단계 4. 만일 서비스 에이전트가 현재 트러스티드 플랫폼에 존재하지 않으면 서비스 제공자에게 요구한다.
단계 5. 서비스 에이전트가 트러스티드 플랫폼으로 이동하여 서비스를 제공한다.

서비스 에이전트는 트러스티드 플랫폼에 상주할 수도 있고, 또는 요청이 있을 때마다 트러스티드 플랫폼으로 이동할 수도 있다. 그리고 이동 에이전트는 서비스 제공자들에 대한 리스트가 미리 정해졌으면, 디렉토리 서비스를 통하여 방문할 트러스티드 플랫폼의 리스트를 얻을 수가 있어서 이동 거리를 최소화할 수도 있다. 만일 방문 순서가 고정되어 있다면 이동 거리는 단지 한 서비스 제공자의 트러스티드 플랫폼이 다수일 경우만 최소화할 수 있게 된다.

4. 결론

이동 에이전트는 온라인 쇼핑, 실시간 장치제어, 분산처리 작업 등에 응용할 수 있는 기술로서 네트워크 트래픽을 감소시킬 수 있고, 특히 이동 장치들과 같이 연결 상태가 좋지 않은 경우에 효과적으로 사용할 수 있는 기술이다. 호스트내에 실행중인 이동 에이전트는 호스트가 반복 실행, DoS, 그리고 변경 등과 같은 공격에 취약한데, 이동 에이전트를 유해 호스트로부터 보호하기 위하여 트러 스티드 플랫폼을 설치하는 것 외에 다른 방법이 아직 까지는 없는 것으로 알려져 있다.

트러스티드 플랫폼의 비용, 설치, 그리고 운영 등의 어려움 등 해결해야 될 여러 가지 문제 점들을 안고 있지만, 트러스티드 플랫폼을 공유함으로써 서비스 제공자들이 쉽게 이동 에이전트를 위한 플랫폼 서비스를 제공하고 신뢰성을 높이고 비용을 줄일 수 있다. 트래픽은 이동 에이전트의 크기와 서비스 제공자측의 에이전트 크기와 네트워크의 토폴로지에 따라 달라질 수 있다. 일반적으로 서비스 제공자 측의 에이전트와 데이터 부분이 이동 에이전트에 비해서 크다고 가정할 때, 네트워크 트래픽은 증가한다고 예측할 수 있다. 이런 경우는 트러스티드 플랫폼에 서비스에이전트의 데이터들을 저장하는 것이 효율적이다.

참고문헌

[1] D. M. Chess, "Security Issues in Mobile Code", in G. Vigna (Ed.), Mobile Agents and Security, Lecture Notes in Computer Science1419, Springer-Verlag, Berlin, 1998, pp1-14
[2] U. G. Wilhelm, S. M. Staamann, L. Buttyan, "A Pessimistic Approach to Trust in Mobile Agent Platforms", IEEE Internet Computing, Sep. 2000 pp40-48.
[3] Bennet Yee, "Using Secure Coprocessor" CMU-CS-94-149, May 1994, School of Computer Science, Carnegie Mellon University
[4] T. Sander, C. Tschudin, "Protecting Mobile Agents against Malicious Hosts", in G. Vigna (Ed.), Mobile Agents and Security, Lecture Notes in Computer Science1419, Springer-Verlag, Berlin, 1998, pp44-60
[5] D. Stinson, "Cryptography: Theory and Practice", CRC, 1995
[6] W. Stallings, "Cryptography and Network Security", Prentice Hall, 1998
[8] A. Young, M. Yung, "Sliding Encryption: A Cryptographic Tool for Mobile Agents", Proc. 4th Int'l Wksp, Fast Software Encryption, FSE
[9] S. Oaks, "JAVA Security", O'Reilly, 1998