이동 에이전트 복제가 가능한 외적 재실행 방지 기법*

김세영, 김연우, 장현수, 엄영익
성균관대학교 정보통신공학부
e-mail:xtrusia@gmail.com, {daroobil, jhs4071, yieom}@ece.skku.ac.kr

Protection Scheme to Clone Mobile Agent against External Replay Attack

Seyeong Kim, Younwoo Kim, Hyunsu Jang, Young Ik Eom
School of Information and Communication Engineering,
Sungkyunkwan University

요 약

이동 에이전트는 플랫폼 간을 이주하며 자기 복제를 통한 작업 분배가 가능하다. 이러한 점에 의해 최근 이동 에이전트는 분산 처리 기반 기술로 각광 받고 있다. 그러나 이러한 에이전트의 이주, 복제 능력은 악의적인 플랫폼과 이동 에이전트의 공격에 대한 약점이 되고 있다. 그 중 재실행 공격은 에이전트의 반복 수행을 통해 에이전트를 공격하는 기법이다. 이에 대한 방지 기법으로 트립 마커를 이용하여 동적인 수행 결정이 가능한 연구가 있다. 그러나 이 기법은 이동 에이전트가 위치한 플랫폼에 따라 이동 에이전트의 복제가 제한 받는다는 단점이 있다. 본 논문에서는 이동 에이전트의 복제가 가능한 외적 재실행 방지 기법을 제안한다. 본 기법은 트립 마커 생성을 담당하는 트립 마커 서버를 두어 플랫폼에 상관없이 유연한 이동 에이전트의 복제가 가능하다. 또한 비대칭키 기법을 이용한 비밀 통신을 통해 재실행 공격으로부터 이동 에이전트를 방어한다.

1. 서론

최근 이동 에이전트를 이용한 분산 컴퓨팅 기술이 주목받고 있다. 이동 에이전트는 다수의 플랫폼을 자율적으로 이동하며, 플랫폼의 자원을 이용하여 작업을 수행한다. 이동 에이전트를 이용한 분산 컴퓨팅 기술은 기존 네트워크 모델에 비해 트래픽 분산 및 병렬처리의 용이성, 비동기식 연산의 효율성 등 상대적으로 분산 컴퓨팅 환경에 유리한 장점들을 가진다. 그러나 이동 에이전트의 작업 수행이 신뢰성을 보장할 수 없는 플랫폼에서 이루어짐으로 인해, 이동 에이전트 복사, 블랙박스 공격, 실행추적에 기인한 코드조작, 이주 경로 조작 등의 공격에 노출되는 문제를 안고 있다. 이러한 보안 문제들을 해결하기 위해 여러 연구들이 진행되고 있다. 알려진 공격 기법 중의 하나인 트립 마커(Trip Marker)를 이용한 외적 재실행 방지 기법은 루프로 진입하는 플랫폼이 신뢰되어야 한다는 제한조건과 이동 에이전트의 복제가 자유롭지 못하다는 문제가 있다[1].

본 논문에서는 이러한 문제점을 해결하고자 트립 마커 서버를 두어 루프로 진입하는 플랫폼이 신뢰되어야 할 제한조건을 완화하고, 이동 에이전트 복제가 가능한 외적 재실행 방지 기법에 대해 제안한다. 본 논문의 2장에서는 트립 마커를 이용한 외적 재실행 방지 기법에 대해 알아보고, 이 연구에서 나타난 문제점을 알아본다. 3장에서는 본 논문에서 제안하는 기법에 대해 설명하고, 4장에서는 제안 기법의 효율성 및 안정성을 증명한다. 마지막으로, 5장에서는 결론 및 향후 연구 과제에 대해서 기술한다.

* 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (IITA-2008-(C1090-0801-0046))

2. 배경지식

2.1 외적 재실행 공격

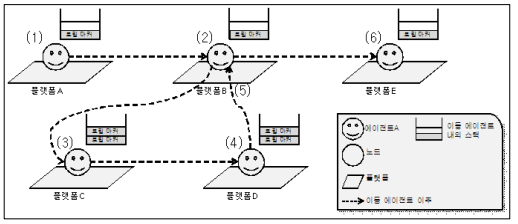
재실행 공격은 네트워크 공격의 한 형태이다. 이는 두 단말이 통신할 때, 한 단말이 다른 단말로 메시지를 계속해서 보냄으로써 이루어지는 공격이다. 이동 에이전트도 이와 같은 재실행 공격에 노출되어 있다. 여러 플랫폼을 이주하면서 주어진 작업을 수행하는 이동 에이전트는 악의적인 플랫폼에 의해 재실행되어지는 공격을 받는다.

재실행 공격은 내적 재실행 공격과 외적 재실행 공격의 형태가 있다. 내적 재실행 공격은 이동 에이전트가 현재 위치한 플랫폼에 의해 이동 에이전트가 반복적으로 재실행되는 공격이다. 외적 재실행 공격은 이동 에이전트 이주 경로 상 루프의 존재와 그 루프를 통해 이동 에이전트가 같은 플랫폼에서 재실행될 것이라는 것을 가정하고 이루어지는 공격이다. 따라서 재실행에 대한 제어나 루프를 거쳐 오는 이동 에이전트를 제어함으로써 외적 재실행을 방지하는 것이 가능하다[2].

외적 재실행 방지 기법으로 이동 에이전트의 최대 수행 횟수 지정 기법과 트립 마커를 이용한 외적 재실행 방지 기법이 있다. 이동 에이전트의 최대 수행 횟수 지정 기법은 이동 에이전트가 플랫폼에 방문하여 수행 할 수 있는 최대 수행 횟수를 사전에 지정해주는 기법으로 정적인 기법이다[3]. 트립 마커를 이용한 재실행 방지 기법은 동적인 기법으로 사전에 정의되지 않고 이동 에이전트가 이주하는 도중에 트립 마커를 이용하여 재실행 여부를 결정한다[1].

2.2 트립 마커를 이용한 외적 재실행 방지 기법

이 기법은 트립 마커를 이용하여 이동 에이전트가 그 실행 여부를 동적으로 결정할 수 있으며 일반적인 구조는 그림 1에서 보인다.



(그림 1) 이주 경로 및 트립 마커

그림 1에서 이동 에이전트가 플랫폼 B에서 루프로 진입하고자 할 때, 먼저 이동 에이전트의 작업을 수행하고 해당 트립 마커를 플랫폼에 저장한다. 그 후 플랫폼은 트립 마커를 생성하여 이동 에이전트 트립 마커 스택에 저장한다. 이 이동 에이전트가 루프를 수행하고, 플랫폼 B를 재방문 할 때, 저장된 트립 마커를 확인하고 수행여부를 결정한다. 이 기법은 이동 에이전트의 최대 수행 횟수 지정 기법을 보완하여, 동적인 결정이 가능한 기법이다.

이동 에이전트의 이주 경로는 순차적인 노드의 집합이다. 최초의 이주 경로는 이동 에이전트가 생산될 때 만들어진다. 이주 경로는 수정되어서는 안 되고, 악의적인 접근으로부터 보호되어야 한다. 이주 경로 보호를 위해 공개키 기반의 전자서명을 이용하며, 현재 노드와 다음 노드에 대한 정보를 암호화하여 이주 경로로 저장하는 기법제안되었다[4].

노드는 플랫폼의 인스턴스이다. 각 노드는 이동 에이전트의 이주가 필요한 다음 플랫폼들을 가리킨다. 이동 에이전트는 하나의 플랫폼을 중복하여 방문하는 것이 가능하기 때문에, 다른 두 노드는 하나의 플랫폼을 가리킬 수 있다. 따라서 각 노드는 하위 노드들을 구성하여 자신만의 루프를 구성한다. 이를 통해 노드는 트리구조를 이룬다. 노드는 부모 노드의 식별자, 부모 플랫폼의 식별자를 갖는다. 이 두 식별자를 통해서 해당 노드가 어느 루프에 속하는지를 확인 할 수 있다.

트립 마커는 Wilhelm에 의해 개념이 제안되고 Garrigues의 연구에까지 이어져 온 핵심 요소이다[5]. 트립 마커는 이동 에이전트 내에 탑재되고, 이동 에이전트가 플랫폼에 이주 시 방문한 플랫폼에 해당 트립 마커를 저장한다. 이동 에이전트는 플랫폼에 저장된 트립 마커를 검사함으로서 중복 방문한 노드를 탐지한다. 또한 이동 에이전트의 작업 수행을 막음으로서 외적 재실행을 방지한다. 트립 마커가 루프 진입 시 새로운 트립 마커를 생성하여 스택에 저장하는데, 이는 이동 에이전트가 각 노드의 루프 수행을 위해 하위 노드에 진입함에 따라 발생하는 노드간 계층 차이를 수용하기 위해서다.

2.3 설계 고려사항

이 기법에서는 이동 에이전트 복제가 제한적이다. 부모 플랫폼만을 신뢰할 수 있는 플랫폼으로 가정하고, 기법의 핵심 요소인 트립 마커 생성이 부모 플랫폼에서만 가능하기 때문이다. 따라서 Garrigues의 기법은 신뢰성 있는 플랫폼으로 이동 에이전트 환경을 구성해야 한다는 제약을 갖는다.

3. 이동 에이전트 복제가 가능한 외적 재실행 방지 기법

본 논문에서 제안하는 기법은 트립 마커 서버를 두어 이동 에이전트 복제가 가능한 외적 재실행 방지 기법을 설계하였다.

3.1 시스템 구조

트립 마커 서버의 신뢰성을 위해 트립 마커 서버는 외부 환경과의 통신은 하지 않는다. 외부 환경과의 통신을 차단하여 본래 주어진 임무만 수행하여, 작업 신뢰성을 향상시킨다.

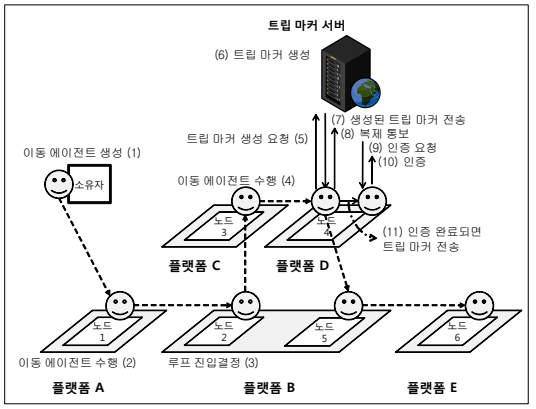
이동 에이전트 복제가 가능한 외적 재실행 방지 기법을 위한 시스템 구조는 플랫폼 및 트립 마커 서버 그리고 이동 에이전트로 구성된다.

이동 에이전트가 이주되어 실행되는 이동 에이전트 플랫폼은 트립 마커를 저장하는 테이블을 유지한다. 이 플랫폼에 방문한 이동 에이전트는 해당 트립 마커를 플랫폼에 저장하게 된다.

트립 마커 서버는 사용자로부터 신뢰 받은 플랫폼으로 트립 마커의 생성을 담당한다. 트립 마커 서버는 이주 대상이 되는 플랫폼 및 이동 에이전트들에 대한 정보를 유지한다. 이 정보에는 플랫폼의 주소, 플랫폼 공개키, 이동 에이전트 식별자, 이동 에이전트 개인키 등이 해당된다.

이동 에이전트는 사용자가 요구한 작업을 수행하기 위한 프로세스이다. 이동 에이전트는 트립 마커의 스택을 가지며, 이동 에이전트가 이동해야 할 이주 경로를 저장한다. 또한 이동 에이전트 내부에는 이동 에이전트 복제를 위해 서버로부터 전송되는 새로운 트립 마커를 임시로 갖고 있을 트립 마커 임시 저장소가 있다.

이동 에이전트가 주어진 여정을 이주하며 자신을 복제할 경우의 예를 그림 2에서 보인다.



(그림 2) 트립 마커 서버가 포함된 이동 에이전트 이주 경로의 예

그림 2에서와 같이, 이동 에이전트가 소유자에 의해서 생성되어 플랫폼 A로 이주하여 작업을 수행한다. 작업을 완료하고 플랫폼 B로 이주하여 첫 방문이라면 작업을 수행한 후 루프에 진입한다. 첫 방문이 아니라면 해당 작업을 완료하고 플랫폼 C로 이주하여 작업을 수행한다. 플랫폼 D로 이주한 후 트립 마커 서버에 트립 마커 생성을 요청한다. 트립 마커 서버는 트립 마커를 생성한 후 요청한 이동 에이전트로 트립 마커를 전송한다. 트립 마커 생성을 요청한 이동 에이전트는 트립 마커 서버로부터 전송된 트립 마커를 임시 저장한 후, 자신을 복제하고 트립 마커 서버에 자신의 복제 사실을 통보한다. 트립 마커 서버는 복제된 이동 에이전트로 인증을 요청하고, 복제된 에이전트는 저장된 복제 식별자를 트립 마커 서버로 보내 인증을 받는다. 인증 완료 후, 트립 마커 서버는 복제된 이동 에이전트의 공개키와 개인키를 생성한다. 해당 개인키는 트립 마커 서버에 저장하고, 공개키는 복제된 이동 에이전트로 전송한다.

3.2 프로토콜

이동 에이전트 복제가 가능한 외적 재실행 방지 기법에서는 공개키 기반의 암호화를 사용한다. 이동 에이전트가 플랫폼 이주 시 방문했던 플랫폼인지 확인을 위해 트립 마커를 확인 한다. 이동 에이전트의 복제를 위해 정의된 프로토콜은 표 1과 같다.

<표 1> 이동 에이전트 복제를 위한 프로토콜

	메시지	전송경로
a	$Msg.Req = E_{K_{EA}}(AID\|NID\|PADDR)$	이동 에이전트 -> 서버
b	$E_{K_{DA}}(E_{K_{EA}}(AID\|NID\|PADDR))$	서버
c	$Msg.Ans = E_{K_{EP}}(AID\|NID)$	서버 -> 이동 에이전트
d	$E_{K_{DP}}(E_{K_{EP}}(AID\|NID))$	이동 에이전트
e	$Msg.CloningID = E_{K_{EA}}(AID_{origin})$	원본 에이전트 -> 복제 에이전트

표 1의 a는 이동 에이전트가 트립 마커 서버로 전송하는 트립 마커 생성 요청 메시지로, 이동 에이전트의 공개키(K_{EA})로 암호화된다.

표 1의 b는 이동 에이전트에서 전송된 트립 마커 생성 요청 메시지를 이동 에이전트의 개인키(K_{DA})로 복호화하는 프로토콜이다.

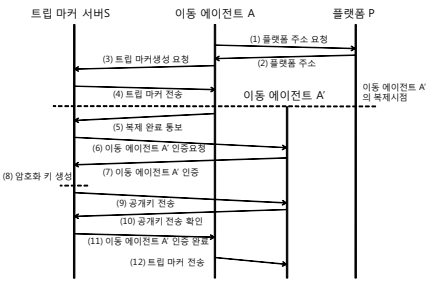
표 1의 c는 트립 마커 서버에서 생성한 트립 마커를 이동 에이전트로 전송하는 메시지로, 이동 에이전트가 위치한 플랫폼의 공개키를 이용해서 암호화 하여 전송한다.

표 1의 d는 트립 마커 서버에서 이동 에이전트로 전송된 메시지를 이동 에이전트가 위치한 플랫폼의 개인키로 복호화 하는 프로토콜이다.

표 1의 e는 이동 에이전트 복제 시 원본 이동 에이전트에서 복제 이동 에이전트로 전송되는 에이전트 식별자로, 원본 이동 에이전트의 공개키로 암호화된다.

4. 시나리오

이동 에이전트 복제 시 이동 에이전트, 플랫폼 그리고 트립 마커 서버 간의 메시지 전송을 그림 3에서 보여준다.



(그림 3) 에이전트 복제 시퀀스 다이어그램

이동 에이전트 A가 일정한 작업 W를 수행하기 위한 목적을 가지고 생성된다. 그 후 이주 경로를 이동하는 중 노드 N에서 에이전트 복제 A′를 생성하고자 한다. 트립 마커 서버는 S이다. 플랫폼 P, P′는 악의적이라 가정한다. 플랫폼 P′ 에는 악의적인 이동 에이전트 A′′가 있다. 이

4.1 플랫폼 주소 요청 공격의 경우

공격자는 이동 에이전트 A가 위치한 플랫폼 P의 주소 요청을 가로채서, 공격자가 원하는 플랫폼 P′의 주소를 답

신에 실어 전송한다. 이 경우 공격자는 트립 마커 서버 S에서 생성된 트립 마커를 플랫폼 P'에 있는 이동 에이전트 A''에서 받기를 기대한다. 그러나 트립 마커 서버에서 전송하는 트립 마커는 이동 에이전트가 위치한 플랫폼에 대한 공개키로 암호화 된다. 전송된 후에 해당 플랫폼에 저장된 개인키로 복호화 하기 때문에, 플랫폼 P'는 전송받은 트립 마커를 이용할 수 없다.

4.2 트립 마커 생성 요청 공격의 경우

공격자는 에이전트 A에서 트립 마커 서버 S로 보내지는 트립 마커 생성 요청을 가로채서 트립 마커를 이용하려 할 수 있다. 하지만 트립 마커로 전송되는 메시지는 이동 에이전트 A의 공개키로 암호화 되어 있어, 트립 마커 서버에 저장된 이동 에이전트 A의 개인키로 복호화 할 수 있다. 따라서 메시지를 가로채더라도 악의적인 플랫폼, 이동 에이전트에서 쓰일 수 없다.

4.3 이동 에이전트 임의 복제 공격의 경우

악의적인 플랫폼이 임의로 이동 에이전트를 복제하려 한다. 공격자에 의해 이동 에이전트가 임의로 생성되었다 하더라도 복제된 이동 에이전트의 트립 마커 생성 요청은 본 이동 에이전트가 하기 때문에 임의의 이동 에이전트는 트립 마커를 받을 수 없다. 또한 이동 에이전트 복제 시 암호화된 복제 식별자를 인증 하는 절차에서 실패하기 때문에, 임의로 복제된 이동 에이전트는 사용이 불가능하다.

5. 결론

Garrigues의 트립 마커를 이용한 외적 재실행 공격에 대한 방지 기법은 재실행 공격 방어에 정적이었던 기존방법을 대체할 동적인 방법을 제시했다. 하지만 그의 연구에서는 부모 플랫폼이 신뢰되어야만 한다는 조건을 두었다. 이 제한사항은 이동 에이전트 환경을 구성할 때 고려해야 할 조건으로서 설계자로 하여금 부담이 되었고, 에이전트 복제가 제한된 플랫폼에서만 가능하다는 단점이 있었다.

이에 본 논문에서는 이동 에이전트 복제가 가능한 외적 재실행 공격 방지 기법을 제안하였다. 사용자로부터 신뢰받는 트립 마커 서버를 두어 반드시 부모 플랫폼이 신뢰되어야 한다는 조건을 제거하였고, 이를 통해 유연한 이동 에이전트 복제를 제공한다.

본 기법은 이동 에이전트가 복제를 할 때 생성되어야 하는 트립 마커에 대해서는 방안을 제시했지만 같이 생성되어야 하는 이주 경로에 대해서는 추후 연구해야 할 사항으로 남겨 놓았다. 복제된 이동 에이전트 이주 경로와 본 이동 에이전트 이주 경로의 동질성은 재실행공격과 같은 성격의 문제를 낳게 된다. 따라서 이주 경로가 어떻게 지정되는지는 중요한 연구 중 하나가 될 것이다.

참고문헌

[1] C. Garrigues, N. Migas, W. Buchanan, S. Robles and J. Borrell, "Protecting Mobile Agents from External Replay Attacks," The Journal of Systems and Software, 2008.

[2] B. Yee, "Monotonicity and Partial Results Protection for Mobile Agents," Proceedings of the 23rd International Conference on Distributed Computing Systems. IEEE Computer Society, pp. 582-591, 2003.

[3] J. Cucurull, J. Ametller, J.A. Ortega-Ruiz, S. Robles, and J. Borell, "Protecting Mobile Agent Loops", Mobility Aware Technologies and Applications. Lecture Notes in Computer Science, Vol. 3744, Springer-Verlag, pp. 74-83, 2005.

[4] J. Mir and J. Borrell, "Protecting Mobile Agent Itineraries," Mobile Agents for Telecommunication Applications (MATA). Lecture Notes in Computer Science, Vol. 2881, pp. 275-285, 2003.

[5] U.G. Wilhelm, S. Staamann and L. Buttyan, "On the Problem of Trust in Mobile Agent Systems," Proceedings of the Symposium on Network and Distributed System Security. Inernet Society, 1998.

[6] T. Aura, "Strategies Against Replay Attacks," Proceedings of the Computer Security Foundations Workshop. IEEE Computer Society, pp. 59-68, 1997.

[7] F. Hohl, "Time Limited Blackbox Security: Protecting Mobile Agents from Malicious Hosts," Mobile Agents and Security. Lecture Notes in Computer Science, Vol. 1419. Springer-Verlag, pp. 92, 1998.

[8] P. Syverson, "A Taxonomy of Replay Attacks," Proceedings of the Computer Security Foundations Workshop. IEEE Computer Society, pp. 131-136, 1994.