상호인증이 가능한 Mobile Agent 보안기법에 관한 연구

서대희, 박희운, 이임영
순천향 대학교 정보기술공학부

A Suggest for Mobile Agent Security in Cross-Certification

Dae-hee Seo, Hee-Un Park, Im-Yeong Lee
Division of Information Technology Engineering. SoonChunHyang University

요약

무선 정보 환경의 변화에 따라 다양한 정보에 대한 풍족감이 요구되면서 양질의 정보를 정확하고 빠르게 선별하여 획득하는 기술이 점점 중요하게 되었으며, 이러한 페러다임의 요구에 Mobile Agent는 중요한 연구가 된다. 본 논문에서는 Mobile Agent가 가져야 할 보안적 요구사항 중 신뢰할 수 있는 Proxy Server를 이용한 인증을 통해 Server와 Mobile Agent간의 안전한 통신 방법을 제시하며, Server나 Mobile Agent의 부정이 발견되었을 경우 DSS(Digital Signature Standard) 방식의 서명값을 이용하여 이를 증명함으로써 보다 안전하고 효율적인 상호인증 방식을 제안하였다.

1. 서 론

Mobile Agent는 여러 가지 다른 실행 환경에서 이용할 수 있는 자동의 소프트웨어 객체이다. 이러한 이동성과 자동성은 영구적인 연결을 필요로 하지 않기 때문에 이기종의 환경에서 수행되는 작업에 유리하다. 그러나 Mobile Agent는 실행의 무결성, 코드의 기밀성, 공개된 기밀 연산, 정당한 인증 등에 대한 보안적 요구사항들이 지적되고 있다.

Mobile Agent 환경에서 고려해 볼 수 있는 보안상의 문제점은 Mobile Agent가 방문한 호스트에 가하는 공격과 호스트가 Mobile Agent에 가하는 공격으로 나누어 볼 수 있다, 이에 대한 연구로는 위조 발견과 위조 방지의 두 가지 측면에서 연구가 진행중에 있다.

먼저 위조 방지 연구는 적극적인 방지와 수동적인 방지로 나눌 수 있으며, 수동적인 방지 연구는 신뢰할만한 실행 환경에서 시스템의 구성과 구조에 대한 연구가 진행 중이다[1].

적극적인 방지에 관한 연구는 Mobile Agent를 호스트의 공격으로부터 보호할 수 있는 기법을 제안하는 것에 초점이 맞추어져 있다.

본 논문에서는 Mobile Agent가 가지는 보안적 요구사항 중 상호인증을 통해 정당한 인증의 보안적 요구사항을 만족하는 보안 기법을 제안하고자 한다.

2. Mobile Agent의 일반적인 인증방법과 문제점

이동 Mobile Agent를 인증하는 방법은 세 가지 방법으로 나눌 수 있다. 첫 번째 인증 방법은 Mobile Agent의 신원을 확인하기 위해 개별 인증 Server를 두어 Mobile Agent ID에 기반을 둔 보안 정책을 강화함으로써 각 Server에 인증을 의존하는 방식이다. 두 번째 인증 방법은 Mobile Agent가 Server에게 자신을 인증 하도록 요구하면 동일한 도메인에 있는 Mobile Agent가 제한된 범위 내에서 인증을 요구하여 인증하는 방식이며, 마지막 인증 방법은 요구되는 보안 정책에 대해 각 Mobile Agent의 신원 증명을 보안영역 관리 Server에게 요구하게 되면, Server가 Mobile Agent의 신원을 다른 영역의 정책 Server에게 증명하도록 요구하는 방법이다. 따라서 Mobile Agent의 문제점을 해결하기 위해서 제안된 방식으로 대리서명 방식, 등급 부여 방식, 정책 프로토콜을 이용하는 방식이 제안되었다.

3. 기존 방식 분석

3.1 대리서명을 이용한 방식

본 방식은 대리서명 기법을 이용하여 안전한 Mobile Agent의 패러다임을 구현하고자 하는 방법으로서 사용자 비밀키의 노출 없이 Mobile Agent의 계산을 호스트에서 수행하도록 하는 기법으로써 악의를 가진 호스트가 취할 수 있는 취약점을 보안하고자 하였다. 그러나 이 방식은 다음과 같은 문제점을 가지고 있다[2].

- 인증 : 호스트와 Mobile Agent간에 상호 인증이 아닌 객체 인증을 통해 이루어지기 때문에 인증의 주체가 되는 객체 자체에 대한 인증이 어렵다는 단점이 있다.
- 요구사항 : Mobile Agent가 가져야할 보안적 요구사항 중 기밀성에 해당하는 비밀값의 계산에 초점을 맞추어 안전성을 강화하고자 하여 Mobile Agent의 전체적 보안적 요구사항에 미흡하다.

- 객체공격 : Mobile Agent나 호스트가 공격자로부터 공격을 받았을 경우 쉽게 객체에 대한 정보의 취득이 가능하여 위장할 수 있는 문제점이 있다.
- 우선순위 : 일정한 필요 정보에 대한 우선순위를 결정하고 시행함에 따라 TTP나 신뢰기관의 참여가 불가피하다.

3.2 등급부여를 통한 방식

본 방식은 Mobile Agent에 등급을 부여하여 등급에 따라 집합을 정의함으로써 Mobile Agent들을 제어하는 방법이다. 등급부여 방식은 악성 Mobile Agent에 대한 명령어의 제한을 통해 Server와 Mobile Agent의 보안을 이루고자 하는 방식이다. 그러나 이 방식은 다음과 같은 문제점을 가지고 있다[3].

- 정책 : 새로운 Mobile Agent에 대한 객관적인 판단을 위한 정책 설정에서 Mobile Agent의 기본 특징이 아닌 시스템 관리자 중심으로 이루어지는 정책이 이루어진다.
- 인증 : 임의의 새로운 Mobile Agent의 인증 단계에서 Mobile Agent가 가져야할 보안적 요구사항 중 기밀성을 중심으로한 인증을 통하여 Server가 공격당했을 경우 공격자에 의해 계속된 악성 Mobile Agent 인증이 가능하다.
- 명령어 제약 : Mobile Agent의 명령어에 대한 제약이 이루어지는 가운데 Mobile Agent간에 상호 인증 명령어 역시 제약되므로 상호 인증이 불가능하다.

3.3 정책 프로토콜을 이용한 방식

본 방식은 안전한 통신 설정을 위해 특정 정책을 기반으로 이루어진 보안 프로토콜을 제시하여 서로 다른 보안 영역간의 Mobile Agent의 협상에 대하여 보안 모델을 제시하였다. 그러나 이 방식은 다음과 같은 문제점을 가지고 있다[4].

- 부하 : 각각의 Mobile Agent, Server간에 통신 부하가 커진다.
- 인증 : 각 Server와 Server, Mobile Agent와 Mobile Agent간에 이루어지는 정보의 교환에서 자료의 무결성에 대한 입증이 안된다는 단점이 있으며 객체간에 상호인증 없이 객체 인증을 통한 악성 Mobile Agent 또는 Server에 대한 부정이 발생할 수 있다.
- 정책 : 정책설정만을 통한 Mobile Agent의 인증으로 정책 Server가 공격당했을 경우 Mobile Agent에 대한 모든 인증은 위협을 받게 된다.

4. 제안 방식

본 논문에서 제안된 방식은 Proxy Server는 안전하다는 가정 하에서 상호인증을 통해 Server와 Mobile Agent간의 안전한 통신을 보장한다.

4.1 Mobile Agent

다음은 Mobile Agent의 시스템 계수이다. 이는 초기에 사용자가 Mobile Agent에게 생성해주는 시스템 계수이다.

x_A : Mobile Agent 비밀키

y_A : Mobile Agent 공개키

α : 사용자가 결정한 난수

r_A : $x^{\alpha} \bmod y$

$ID(A)$: Mobile Agent 정보

h_A : Mobile Agent의 안전한 해쉬값

Mobile Agent는 초기 생성시 사용자가 생성해준 x_A, y_A, α, r_A를 기본 코드값으로 가지고 있다. Mobile Agent가 Server에 접속을 요구함과 동시에 접속하고자 하는 Server의 Proxy Server에게 사용자가 결정한 난수 α를 이용하여 계산한 r_A와 $ID(A)$, 그리고 h 값을 다음과 같이 생성하여 Proxy Server에 전송한다.

$$h_A = H(r_A \parallel ID(A))$$

4.2 Proxy Server

다음은 안전하다고 가정한 Proxy Server의 시스템 계수이다.

x_P : Proxy Server가 생성한 소수 ($512 \le L \le 1024$, $2^{L-1} < x_P < 2^L$을 만족하는 소수)

y_P : $(x_P - 1)$의 소수

ε : $\varepsilon^{(x_P-1)/y_p}$를 만족하는 $1 < \varepsilon < (x_P - 1)$ 정수

p_p : Proxy Server 공개키

q_p : Proxy Server 비밀키

k, β : Proxy Server에서 생성한 난수

R_A, S_A : DSS 서명을 위한 서명계수

r_B : Proxy Server가 생성한 Y_P의 구성요소

Y_P : Proxy Server가 생성한 검증값

h_P : Proxy Server가 생성한 안전한 해쉬값

Proxy Server는 내부 난수 생성기를 통해 난수 k와 β를 생성한다. 이를 통해 Proxy Server는 Mobile Agent에게서 전송된 r_A와 Proxy Server에서 생성한 난수 β를 이용하여 r_B를 다음과 같은 방법으로 생성하며, ε를 이용하여 서명을 위한 g_P를 생성한다.

$$g_P = \varepsilon^{(x_P-1)/y_p} \bmod y_P$$

$$r_B = r_A^{\beta} \bmod p_p$$

또한 서명용 알고리즘인 DSS 방식을 이용하여 Mobile Agent 서명용 계수인 R_A, S_A를 다음과 같은 방법으로 생성한다.

$$R_A = (g_P^{k} \bmod x_P) \bmod y_P$$

$$S_A = k^{-1}(H(ID(A)) + q_P R_A) \bmod y_P$$

Proxy Server는 Mobile Agent를 위한 R_A, S_A 값을 생성한 후 Mobile Agent와 Server에게 공통적으로 전송할 Y_P를 다음과 같이 생

성하게 된다.

$$Y_P=(r_A \cdot r_B \cdot r_H)^{\beta} \bmod p_p$$

4.3 Server
다음은 Server의 시스템 계수이다.

x_H : Server가 생성한 소수 ($512 \le L \le 1024$, $2^{L-1} < x_H < 2^L$을 만족하는 소수)
y_H : ($x_H - 1$)의 소수
ζ : $\zeta^{(x_P-1)/y_p}$를 만족하는 $1 < \zeta < (x_P-1)$ 정수
p_H : Server 공개키
q_H : Server 비밀키
σ : Server가 생성한 난수
R_H : DSS 서명을 위한 서명계수
S_H : DSS 서명을 위한 서명계수
$ID(H)$: Server 정보
h_H : Server의 안전한 해쉬값

Mobile Agent의 접속 요청을 Proxy Server가 인지하였을 경우, Proxy Server는 접속 요청을 한 Mobile Agent의 접속 요구 메시지를 Proxy Server가 대신하여 Server에게 전송한다. Server는 Proxy Server의 요청이 있을 경우 난수 σ를 이용하여 r_H값을 다음과 같이 생성하여 Proxy Server에게 전송하며, DSS서명을 위해 ζ를 이용하여 서명을 위한 g_H를 생성한다.

$$r_H = p_H^{\sigma} \bmod x_H$$
$$g_H = \zeta^{(x_H-1)/y_H} \bmod y_H$$

Server는 DSS서명 알고리즘을 이용하여 다음과 같이 R_H, S_H값을 생성한다.

$$R_H=(g_H^{\sigma} \bmod x_H) \bmod y_H$$
$$S_H=\sigma^{-1}(H(ID(H))+q_H R_H) \bmod y_H$$

4.4 Handshake 단계
본 논문에서 제안된 상호인증 방식을 단계별로 분석하면 다음과 같다.

[단계 1] (**Mobile Agnet 접속 요구단계**)
① *접속 요구 Server의 해당 Proxy Server에서 Mobile Agent와 Server의 정보 요구*
Mobile Agent : r_A, $ID(A)$, $h_A(r_A \| ID(A))$
Server : r_H, $ID(H)$, $h_H(r_H \| ID(H))$, R_H, S_H

[단계 2] (**상호인증과 서명값 생성단계**)
② *Proxy Server에서 상호인증을 위한 Y_P 생성 및 Mobile Agnet 서명값 생성 단계*
Proxy Server :

$$Y_P=(r_A \cdot r_B \cdot r_H)^{\beta} \bmod p_p$$

Mobile Agent를 위한 서명값 생성

$$R_A=(g_P^{k} \bmod x_P) \bmod y_P$$
$$S_A=k^{-1}(H(ID(A)))+q_P R_A) \bmod y_P$$

[단계 3] (**전송단계**)
③ *Proxy Server가 Mobile Agent 상호인증 계수 전송단계*
전송되는 상호 인증 계수

$$r_A, R_A, S_A, Y_P, h_P(r_A \| R_A \| S_A \| Y_P)$$

④ *Proxy Server가 Server에게 상호인증 계수 전송단계*
Server의 공개키고 암호화하여 전송

$$E_{P_H}(ID(H) \| Y_P), h_P(Y_P \| ID(H))$$

[단계 4] (**상호인증 단계**)
⑤ *Mobile Agent와 Server의 상호인증 단계*
Mobile Agent → Sever

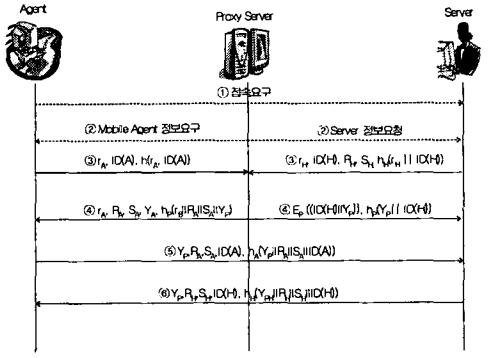
$$Y_P, R_A, S_A, ID(A), h_A(Y_P \| R_A \| S_A \| ID(A))$$

Server → Mobile Agent

$$Y_P, R_H, S_H, ID(S), h_H(Y_H \| R_H \| S_H \| ID(S))$$

[단계 5] (**검증 단계**)
⑥ *Mobile Agent와 Server의 상호 검증 단계*
Mobile Agent
Mobile Agent $Y_P \stackrel{?}{=}$ Server Y_P
Server
Server $Y_P \stackrel{?}{=}$ Mobile Agent Y_P
⑦ *Proxy Server의 부정확인*
Proxy Server가 각 객체에게 요구하는 정보
Mobile Agent → Proxy Server : R_A, S_A , Y_P
Server → Proxy Server : R_H, S_H , Y_P
Proxy Server $Y_P \stackrel{?}{=} Y_P \stackrel{?}{=} Y_P$ 검증

Proxy Server, Mobile Agent, Server와의 전체적인 Handshake 과정은 (그림 1)과 같다.



(그림 1) Proxy Server를 이용한 Mobile Agent와 Server의 상호인증

본 논문에서는 안전한 Proxy Server를 이용하여 Mobile Agent와 Server의 상호인증을 제안하였다. 제안 방식은 Proxy Server는 안전하다는 전제로 제안된 방식이며 이를 바탕으로 Moblie Agent가 가져야할 보안사항을 분석하면 다음과 같다.

① 무결성 : Mobile Agent는 Server와 상호인증을 통해 악의적인 서버의 공격으로부터 사전에 이를 인지할 수 있다.
② 기밀성 : Mobile Agent가 수행하고자 할 때 Proxy Server에서 계산된 값을 Agent에 포함되므로 기밀성 유지할 수 있다.
③ 기밀연산 : Mobile Agent는 원격지에서 사용자의 초기 비밀키를 노출시키지 않고 Y_P값으로 상호인증을 하며 DSS 서명값인 R_A, S_A값으로 디지털 서명값을 수행한다.
④ 정책 : 제안 방식은 특정한 정책이 필요없이 Proxy Server를 통한 상호인증이 가능하다.
⑤ TTP의 의존도 : 제안 방식에서 Proxy Server는 객체간의 상호인증을 통한 믿을 수 있는 객체로 설정되며, 검증을 위한 신뢰기관으로 제안되었으므로 TTP의 의존도는 매우 높다.
⑥ 인증 : Mobile Agent와 Server는 Proxy Server에게서 전송된 Y_P로 상호인증 할 수 있으며 부정이 발생시 Proxy Server를 통해 Y_P와 DSS서명값을 검증 받을 수 있다.

이상의 제안된 방식의 분석을 통해 기존 방식과 비교 분석하면 <표 1>과 같이 정리할 수 있다.

<표 1> 기존 방식과 제안방식 비교분석

	대리서명	등급부여	프로토콜 방식	제안방식
무결성	×	×	×	○
기밀성	○	○	×	○
기밀 연산	×	×	×	○
정책	×	○	○	×
TTP 의존도	높다	높다	높다	높다
인증	객체인증	객체인증	객체인증	상호인증

5. 결론

본 논문에서는 신뢰할 수 있는 Proxy Server를 이용하여 Mobile Agent와 Server 사이에 상호인증을 통한 안전한 Handshake 과정을 제안하였다.

Mobile Agent와 Server와의 상호인증은 각 객체간에 신뢰성을 줄 뿐만 아니라 부정이 있는 객체를 증명함으로써 상호 보안적인 관계를 유지시킬 수 있다. 따라서 본 논문에서 제안한 방식은 Mobile 환경에서 Agent를 통해 사용자가 안전하게 정보를 제공받을 수 있을 뿐만 아니라 Mobile 환경의 여러 다른 응용 서비스에 적용 가능하리라 사료된다.

6. 참고 문헌

[1]. 장병탁, 이종우, 서용우 "학습 에이전트", 정보과학회지 제 18권 5호, p26, 서울대학교, 2000. 5.
[2]. 김희선, 백준상, 이병천, 김광조, "대리서명을 이용한 모바일 에이전트의 안전성 강화 방법 (Enhancing Security of Mobile Agent using Proxy Signature)", KIISC 종합학술발표회(CISC2000), 종합학술발표 논문집 Vol. 10 No. 1, pp.424~437, 성균관대학교, 2000. 11.
[3]. 임용성, 장덕성, 정홍 "명령어 등급 부여를 통한 에이전트의 불법행위 방지에 관한 연구", 정보처리학회 논문집 VOL7 NO 8S, pp.2641~2649, 2000.8.
[4]. 김영덕, 신동명, 최용락 "SPS를 기반으로한 보안영역간 이동 에이전트 인증 협상 모델", 한국통신정보보호학회 충청지부, pp.241~254, 대전대학교, 2000.11.
[5]. 최용락, 소우영, 이재광, 이임영 "통신망 정보 보호", 도서출판 그린, 1997.2.
[6]. 이만영, 김지홍, 류재철, 송유진, 염흥렬, 이임영 "전자상거래 보안 기술", 생능출판사 1999.8.
[7]. 최용락, 소우영, 이재광, 이임영 "컴퓨터 통신 보안", 도서출판 그린, 2001.2.
[8]. Alfred J. Menezes, Paul C.van Oorschot, Scott A. Vanstone "HANDBOOK of APPLOED CRYPTOGRAPHY", CRC, 1996.11.