실행상태 정보 보호를 위한 디지털 다중서명 기반 이동에이전트 보안시스템의 설계

김 경 , 정 일 용
조 선 대 학 교 전 자 계 산 학 과
liprince@naver.com , iyc@mina.chosun.ac.kr

The Design of Secure Mobile Agent System based on Digital Multi-Signature Scheme for protecting agent state

Kyoung KIM, Il-Yong Chung
Dept. of Computer Science, Chosun University

요 약

Mobile Agent System은 최근 각광받기 시작한 새로운 컴퓨팅 기법으로서 서버의 과부하를 줄이고, 네트워크 이벤트의 발생을 줄이면서 필요한 서비스를 사용자에게 지원하기 위한 해결책으로 연구되고 있다. 이동에이전트시스템은 해결해야할 많은 과제들이 있으나 그 중에서 주요한 문제로 인식되고 있는 것이 보안 문제이다. 본 논문은 이동에이전트 시스템의 보안 고려사항을 분석하고, 공개키 기반의 암호시스템과 일방향함수를 사용하는 디지털 다중서명 기법을 적용하여 이동에이전트 시스템의 보안구조를 제안한다.

1. 서 론

최근 컴퓨터시스템과 네트워크시스템 분야에서는 '대리인'의 개념을 이용하여 사용자의 시스템 사용을 보다 쉽도록 하는 연구가 각 분야에서 활발히 연구되고 있다. 그 중에서 서버의 과부하를 줄이고, 네트워크 이벤트의 발생을 줄이면서 필요한 서비스를 사용자에게 지원하기 위한 통신 응용 프로그램들은 데이터와 코드를 결합한 객체의 이동 즉, 이동에이전트를 사용하고 있다. 이동 에이전트는 현재 객체의 상태와 실행 가능한 코드를 포함하는 객체로서 분산환경에서 특정서비스를 제공하기 위하여 서로 통신할 필요가 있는 호스트간 직접 이동을 통하여 실행되는 객체[1]로서 자신의 홈 플랫폼에서 출발하여 주어진 일을 완수할 때까지 네트워크상의 플랫폼들을 이동한다. 이동에이전트는 전자상거래, 그룹공동작업, 이벤트 모니터링, 작업흐름 자동화, 정보검색, 망관리 및 이동컴퓨팅 등에 이용될 수 있어 그 활용 범위가 매우 넓다. 그러므로 여러 응용에 안전하게 이동에이전트 시스템을 이용하기 위해서는 보다 효과적으로 에이전트 시스템을 보호하는 방법이 요구되고 있다. 본 논문에서는 이동에이전트 시스템의 보안문제 중 에이전트가 에이전트 서버를 이동시 서버로부터 받을 수 있는 공격에 대응할 수 있도록 순차 다중서명 방식의 하나인 'Okamoto 다중 서명 방식'[12]을 이용하여 에이전트 보안시스템을 제안하고자 한다.

2. 이동에이전트 시스템의 보안 고려요소

2.1 이동에이전트의 보호문제

이동에이전트 시스템은 분산 어플리케이션의 구성에 유연한 환경을 제공해 주지만 심각한 보안 문제를 야기한다. 첫째는 불완전한 통신채널을 이용하는 것에 대한 기본적인 보안 문제가 있으며, 둘째는 에이전트가 행하는 불법적인 행위나 공격에 대한 호스트 컴퓨터와 에이전트 서버의 보호문제이고, 셋째는 에이전트 서버가 행하는 에이전트공격에 대한 보안 문제이다.

에이전트가 호스트 컴퓨터를 공격하는 경우 디지털서명을 이용한 에이전트의 인증과 접근레벨에 따른 접근제어[3]로서 해결될 수 있다. 또한 JAVA 보안모델이 지원하는 보안 메카니즘을 사용하면 서버 보안 문제를 해결할 수 있다. 하지만 에이전트는 이동한 서버에게 모든 것이 노출되고 서버는 에이전트의 코드나 상태를 변경함으로서 에이전트의 행동을 변경/방해할 수 있다. 에이전트의 실행상태는 항상 변화하기 때문에 에이전트 서버의 불법적인 변경행위의 공격에 매우 취약하게 된다. 때문에 에이전트 서버의 공격에 대한 에이전트의 보호는 해결하기 어려운 문제이다. 이러한 공격을 탐지하기 위해서는 에이전트의 모든 실행상태를 생성자에게 전달하고 생성자가 이것을 확인할 수 있는 방법이 필요하다. 이동 에이전트의 보호를 위한 연구[4,5]가 이루어져 왔으나 이러한 방법들은 네트워크 자원을 남용하거나 실행시간이 많이 걸리고 비용이 높다는 지적을 받고 있는 실정이다. 또한 디지털서명과 감사도구를 이용한 보안구조[6]를 제안하였으나 에이전트의 실행 도중에 발생하는 변경행위에 대하여 즉각 생성자에 보고하지 않음으로서 불필요한 overhead를 갖을 수 있다. 그러므로 여러 응용에 안전하게 이동에이전트 시스템을 이용하기 위해서는 보다 효과적으로 에이전트를 보호하는 방법이 요구된다.

3. Digital Multi-signature

3.1 디지털 서명

디지털 서명은 전자적인 정보를 이용하여 디지털 메시지에 서명하는 것으로 메시지 인증과 사용자 인증[7]을 할 수 있어야 한다. 디지털 서명 방식에는 RSA 암호 시스템을 이용하는 방식, 이산대수 문제의 어려움에 근거를 둔 ElGamal 암호를 이용하는 방식, 변형된 Knapsack문제를 이용한 방식 그리고 ID를 이용한 디지털 서명 방식 등이 있다. 이와 같은 디지탈 서명의 기능은 전송되는 정보의 불법변경 여부를 판별할 수 있게 해주는 무결성(Integrity) 기능, 사용자 인증 기능, 정보의 송신이나 수신을 부인할 수 없게 하는 부인방지 (non-repudiation) 기능 등에서 필수적인 도구로서 이용되고 있다. 지금까지 디지털 서명은 한사람이 어떤 메시지에 전자적으로 서명하는 것으로써 단순 서명(single signature)이다. 디지털 다중서명(Digital Multi-signature)이란 동일한 메시지에 대해 여러사람이 전자적으로 서명하는 것을 말한다. 기존의 디지털 다중 서명 방식으로는 두 개의 큰 소수와 각 서명자의 직위에 따른 작은 소수의 곱을 이용하여 RSA 방법을 확대 적용한 Itakura-Nakamura방법, RSA 방식과 같은 전단사 공개키 암호 시스템과 단방향함수를 이용한 Okamoto방법, Fiat-Shmir 서명 방식에 근거하여 만들어진 Brickell-Lee-Yacobi방법, Ohta-Okamoto 방법[8] 등이 연구되어있다. 이 중 RSA 공개키 방식으로 운용되며 서명순서에 제약이 없고, 중간서명자가 검증이 가능한 Okamoto 방법을 본 논문에 적용한다.

4. 다중서명을 이용한 에이전트의 보호

에이전트는 실행 코드와 실행 상태를 가진다. 또한 이동에이전트 코드는 모든 호스트에서 동일한 형식으로 동작되어야 하고, 직접 해석되거나 다시 컴파일 되지 않고 수행될 수 있는 이식성이 있는 중간언어이어야 한다. 인터프리터는 에이전트의 호스트 자원에 대한 접근을 제어하면서 에이전트를 실행시킨다. 에이전트 서버로부터 받은 에이전트의 코드와 상태에 의해 인터프리터는 에이전트를 실행시킨다. 본 논문의 보안구조시스템은 에이전트를 구동시키는 인터프리터가 신뢰성 있게 구현되어 에이전트 서버에서 구동되고 있다고 가정한다. 실행 코드는 에이전트 생성시 만들어지고 소멸할 때까지 변하지 않는다. 따라서 에이전트 실행 코드는 생성자의 디지털 서명을 통해서 코드가 변경되었는지에 대한 확인을 할 수 있다. 실행 상태는 에이전트가 실행되면서 계속 변화하기 때문에 각 에이전트 서버는 에이전트 실행 상태에 대한 디지털 서명을 함으로써 에이전트 실행 상태 변경에 대한 부인 봉쇄를 이룰 수 있다.

4.1.Notation

AS_i : i번째 에이전트 서버의 ID
$Info_i$: AS_i에서 에이전트의 실행 과정 상태정보
C_i : 에이전트가 경유한 i번째 에이전트 서버까지의 갯수
h_i : AS_i의 공개된 단방향 함수
Ee_i : 키 e_i에 의한 공개키 암호함수 (AS_i의 공개키)
Dd_i : 키 d_i에 의한 공개키 복호함수 (AS_i의 비밀키)
RS_n : 수신 부인봉쇄를 위한 n번째 에이전트 서버의 응답
X_n : 서명자 n의 평문과 암호문의 유한 집합
$|N|$: N의 비트길이
$[S]^L$: S의 ($|S|-L$)개의 최상위 비트. 즉, $|[S]^L| = |S| - L$
$[S]_L$: S의 L개의 최하위 비트. 즉, $|[S]_L| = L$
$^L\{S\}$: 상위 ($L-|S|$)개의 '0'비트 패딩을 갖는 S.
$||$: 연접(concatenation)

4.2 알고리즘

(1) 에이전트 서버의 등록

에이전트 서버 i는 RSA 알고리즘에 의해 공개키 e_i와 비밀키 d_i를 생성한 후, 공개키 e_i와 단방향 해쉬함수 $h_i : X_i X_i X_i \ldots X_i \rightarrow X_i$를 공개하고 비밀키 d_i를 자신이 보관한다. 또한 자신의 ID인 AS_i를 Directory server에 등록한다.

AS_i → Directory server : e_i, h_i, AS_i

(2) AS_1의 서명발생

에이전트 AS_1에 있었다는 사실을 부인하지 못하도록 다음과 같이 디지털 서명한다. 만일 각각의 에이전트 서버에서의 에이전트 실행과정의 상태를 다른 에이전트 서버에게 노출되지 않고 에이전트 생성자만이 확인할 수 있도록 해야하는 경우에는 이 에이전트 서버의 실행과정의 상태정보를 에이전트 생성자의 공개키로 암호화한다

AS_1 → AS_2 : $E_{e_2}(S_1, M_1)$, $E_{e_2}(AS_1)$

$S_1 : D_{d_1}(h_1(M_1))$

$M_1 : Info_1 || C_1$

(3) AS_2의 검증

AS_1으로부터 받은 (S_1, M_1)이 다음 식을 만족하면 AS_1의 서명 메시지가 유효한 것으로 간주한다.

$$E_{e_1}(S_1) = h_1(M_1)$$

AS_1의 서명 메시지가 유효한 것으로 검증이 되면, AS_2는 AS_1에게 자신이 AS_1의 서명과 메시지를 받았다는 응답을 자신의 비밀키로 서명하여 전송한다.

AS_2 → AS_1 : $E_{e_1}(RS_2)$

$RS_2 : D_{d_2}(S_1, M_1, 'Received')$

(4) AS_2의 서명

AS_2는 에이전트를 실행하여 얻은 상태정보에 다음과 같이 서명한다. C_2는 AS_1으로부터 넘겨받은 에이전트 서버의 ID갯수에 1을 더하여 구하며, 이것은 에이전트를 실행한 에이전트 서버가 이전 에이전트 서버들에서 실행된 상태정보와 에이전트 식별자를 제거하는 것을 체크하기 위한 카운트이다.

AS_2 → AS_3 : $E_{e_3}(S_2, M_2)$, $E_{e_3}(AS_1, AS_2)$

$|X_2| > |X_1|$ 이면

$S_2 : D_{d_2}(^{|X_2|}\{S_1\} || h_2(Info_2 || C_2))$

$M_2 : M_1 || Info_2 || C_2$, $C_2 : C_1 + 1$

그렇지 않으면

$S_2 : D_{d_2}(^{|X_2|}\{[S_1]_{|X_2|-1}\} \| h_2(Info_2 \| C_2))$

$M_2 : M_1 \| [S_1]^{|X_2|-1} \| Info_2 \| C_2 \ , \ C_2 : C_1 + 1$

(5) 다중서명 검증

n번째 에이전트 서버는 (S_{n-1}, M_{n-1})을 수신하면 다음과 같이 검증한다. 검증과정에서 문제가 발생하는 경우 즉각 홈으로 통보함으로써 에이전트의 실행을 중지할 수 있도록 한다. 공개키 e_i(i=1,2,....,n-1)를 이용하여 다중서명 메시지를 점검한다. 에이전트 서버의 순서는 서명메시지에 첨부된 에이전트 서버의 식별자(AS_1,.....,AS_{n-1})에 의하여 표시된다. 각각의 S_i'와 M_i'가 검증식을 만족하면 i번째 에이전트 서버의 서명과 실행상태 정보에 대한 무결성이 증명되며 다중서명 메시지는 유효 한 것으로 간주한다. 여기서 $M_{n-1}' = M_{n-1}$ 이고 $S_{n-1}' = S_{n-1}$이다.

$[E_{e_i}(S_i')]_{|X_i|} = h_i([M_i']_{|(Info_i \| C_i)|})$

$|X_i| > |X_{i-1}|$ 이면

$S_{i-1}' : [[E_{e_i}(S_i')]^{|X_i|}]_{|X_{i-1}|}$

$M_{i-1}' : [M_i']^{|(Info_i \| C_i)|}$

그렇지 않으면

$S_{i-1}' : [[M_i']^{|(Info_i \| C_i)|}]_{|X_{i-1}|-|X_i|+1} \|$

$[[E_{e_i}(S_i')]^{|X_i|}]_{|X_i|-1}$

$M_{i-1}' : [[M_i']^{|(Info_i \| C_i)|}]_{|X_{i-1}|-|X_i|+1}$

에이전트를 수신 받은 AS_n이 에이전트를 정지시키는 공격 행위에 대하여 AS_n이 에이전트를 받았다는 증명을 AS_{n-1}에게 제공하도록 함으로써 에이전트 turnarround가 없을 경우 문제를 발생시킨 AS를 찾도록한다. 즉, 에이전트를 보낸 AS_{n-1}은 AS_n으로부터 다음과 같은 RS_n을 받아두도록 함으로써 가능하며, 수신메세지를 받지 못하는 경우는 홈에게 AS_n이 문제가 있음을 알리도록한다.

$\mathbf{AS_n \rightarrow AS_{n-1}} : E_{e_{n-1}}(RS_n)$

$RS_n : D_{d_n}(S_{n-1}, M_{n-1}, \text{'Received'})$

(6) AS_n의 서명

$\mathbf{AS_n \rightarrow AS_{n+1}} : E_{e_{n+1}}(S_n, M_n), \ E_{e_2}(AS_1, AS_2, \ldots, AS_n)$

$|X_n| > |X_{n-1}|$ 이면

$S_n : D_{d_n}(^{|X_n|}\{S_{n-1}\} \| h_n(Info_n \| C_n))$

$M_n : M_{n-1} \| Info_n \| C_n \ , \ C_n : C_{n-1} + 1$

그렇지 않으면

$S_n : D_{d_n}(^{|X_n|}\{[S_{n-1}]_{|X_n|-1}\} \| h_n(Info_n \| C_n))$

$M_n : M_{n-1} \| [S_{n-1}]^{|X_n|-1} \| Info_n \| C_n$

$C_n : C_{n-1} + 1$

만약 서명이 마지막 서버에서 이루어졌다면 (S_n, M_n)과 (AS_1, AS_2,...AS_n)을 홈으로 보낸다.

5. 결론

본 논문에서는 최근 많은 연구가 진행되고 있는 에이전트 시스템 중에서 이동 에이전트 시스템에 대하여 살펴보고 이동 에이전트 시스템에서 가장 큰 문제로 지적되고 있는 보안문제에 대하여 기술되었다. 이동 에이전트 시스템에서 야기될 수 있는 보안 문제 중 이동 에이전트가 이동 중 또는 다른 호스트로 이동 한 후 악의적인 호스트로부터 받을 수 있는 공격에 대한 보안으로 공개키 기술에 기반을 둔 디지털 다중서명 방법을 적용하였다. 제안된 이동 에이전트 보안 구조의 장점은 다음과 같다.

첫째, 각 에이전트 서버에 의한 디지털 다중 서명을 이용하여 에이전트 실행 상태 변화를 각 서버마다 검사토록 함으로써 문제 발생시에 즉각 발견할 수 있도록 하였다.

둘째, 사용된 다중 서명은 Okamoto 다중 서명 방식에 근거하므로, 서명순서의 제약이 없다. 즉, 이동 에이전트의 경로 배정 문제에 있어서 이동 에이전트의 생성자가 미리 경로를 배정하여 이동 에이전트는 그 경로를 따라서만 이동하는 경우 뿐만 아니라, 이동 에이전트가 갖는 기본적인 특성인 자율성과 지능을 바탕으로 이동 에이전트가 직접 이동할 에이전트 서버를 결정하여 이동하는 경우에도 적용될 수 있다.

셋째, 공개키 암호시스템과 단방향 해쉬 함수를 사용함으써 정보를 보호하는데 기본적으로 제공되어야 할 기밀성, 인증, 무결성, 부인봉쇄 등의 기능을 제공할 수 있다.

그러나 기본적으로 적용된 전단사 공개키 암호시스템과 일방향함수(one-way function)를 이용한 Okamoto 방법이 RSA방법에 기초하기 때문에 서명시 계산량이 많아져 처리속도가 느리다는 단점이 있다. 향후 좀 더 수행 능력이 뛰어난 방법으로 실행 상태정보에 대한 보안 문제를 해결하는 구조를 연구하고자 한다.

6. 참고문헌

[1] 박지은,김상욱, "이동 에이전트를 지원하는 프로그래밍어", 정보처리학회지, 제 4권, 제 5호, pp.34 ~ 41, 1997

[2] T.Okamoto, "A Digital Multisignature Scheme Using Bijective Public-key Crypto-systems", ACM Trans. on Comp.systems Vol 6, No 8, pp.432~441, 1988.

[3] David Chess, Benjamin Grosof,Colin Harrison, David Levine,Colin Paris, Gene Tsudik, "Itinerant Agents for Mobile Computing"/Technical report, IBM Research Division-T.J. Watson Research Center, 1995.

[4] Bennet S.Yee, "A Sanctuary for Mobile Agents", DARPA Workshop on Foundations for Secure Mobile Code Workshop, pp.26~28, 1997.

[5] Giovanni Vigna, "Protecting Mobile Agents through Trac ing" , ECOOP Workshop, 1997.

[6] 백주성, 이동익, "디지털 서명과 감사 도구(Audit trail)를 이용한 이동 에이전트의 보호", 한국정보과학회 학술발표논문집, 제 24권, 제2호, pp.419~422, 1997.

[7] 현대암호학, 한국전자통신연구소, pp.157~170, 1991.

[8] 강창구, "디지탈 다중서명 방식과 응용에 관한 연구", 공학박사학위논문, 충남대학교, pp.41~43, 1993.