MH 환경에서의 이동 에이전트를 이용한 안전한 구매 프로토콜

*허 세 민°, *최 영 근, *김 순 자
*경북대학교 전자전기공학부

A Study of Secure Purchase Protocol with Mobile Agent in Multiple-Hop Environments

*Se-Min Heo°, *Yeong-Geun Choe *Soon-Ja Kim
*School of Electronics and Electrical Eng., Kyungpook National University
*{urimini°, ind}@palgong.knu.ac.kr

요 약

이동 에이전트는 자동으로 여러 새로운 호스트를 이동하여 사용자가 원하는 정보를 수집하거나 물품에 대한 구매 등을 할 수 있게 해주는 프로그램이다. 그리고 이러한 이동 에이전트를 사용함으로서 인터넷 환경, 특히 전자상거래에서의 많은 이익을 가져올 수 있다. 하지만 포괄적인 security framework의 부족으로 이동 에이전트의 보급과 사용에 많은 제한이 있다. 즉 악의적인 호스트에 의해서 에이전트의 state나 code가 위조될 수 있고 사용자에게 피해를 주게된다. 본 논문에서는 Multiple-Hop 환경에서의 이동 에이전트를 이용한 데이터 검색과 구매가 위조 및 변경되는 것을 감지하고 방지할 수 있는 프로토콜을 제안하였다.

1. 서론

Corradi 등은 이동 에이전트의 무결성을 보장하기 위해서 TTP(trusted third party)라는 프로토콜과 MH(multiple hop)라는 프로토콜을 제안하였다[1,2].

TTP 프로토콜은 이동 에이전트가 각 사이트를 방문할 때마다 어떤 신뢰 기관을 방문하여서 에이전트의 무결성을 검증할 수 있도록 설계한 프로토콜이다.

하지만 에이전트가 매번마다 신뢰기관을 방문해야하기 때문에 그에 따른 비용이 많이 든다. 이를 개선한 것이 MH 프로토콜이다.

이 프로토콜은 TTP 프로토콜과는 달리 에이전트가 TTP와의 상호작용 없이 네트워크를 자유롭게 이동할 수 있도록 설계된 것이다. 각 사이트들은 에이전트에게 어떤 proof를 제공해야하며 이러한 proof들이 chain형식으로 에이전트에 저장이 된다. proof는 이전 사이트에서 계산된 것과 암호학적으로 연결되어있기 때문에 어떤 악의적인 호스트가 proof를 위조할 수가 없다.

결국 에이전트가 owner에게 돌아왔을 때 에이전트에 저장되어있는 proof chain의 검증을 통해 에이전트의 무결성을 증명할 수 있다. 비록 MH 프로토콜이 TTP 프로토콜처럼 중간에 어떤 *check point*가 없기 때문에 에이전트의 위조를 즉시 알 수 없다는 단점이 있지만 TTP 프로토콜에 비하여 비용이 적게 든다는 장점이 있다.

Kotzanikolaou 등은 Sander와 Tschudin이 제안한 undetachable signature의 문제점들을 해결하면서 사용자와 서버사이에서 안전한 거래가 일어날 수 있는 방법을 제시하고 있으며, 그로 인하여 에이전트가 사용자의 개인키를 노출시키지 않고 서버에 있는 메시지에 안전히 서명을 할 수 있는 방법을 제안하였다[3].

본 논문에서는 앞에서 제시한 방법들을 분석하고 그 단점들을 보안하여 MH 환경에서 에이전트의 무결성을 지키며 사용자와 서버사이에 안전한 거래가 일어날 수 있는 방법을 제안한다.

논문의 구성은 2절에서는 MH 프로토콜에 대한 분석, 3절에서는 사용자의 안전한 서명을 위한 Kotzanikolaou 등이 제안한 방법과 프로토콜 분석, 4절에서는 제안하는 프로토콜, 5절에서는 결론을 맺는다.

2. MH Protocol

에이전트는 세 부분으로 구성되어 있다. 그 구성요소를 보면 첫째가 CID(code and initialization data), 둘째가 AD(application data), 셋째가 PD(protected data) 이다. CID에는 에이전트의 실행 코드, 사용자의 Identity, 사용자의 질의 등을 포함하는 req_C가 포함되며 내용이 변경될 수 없도록 사용자의 개인키로 서명되어 있다. AD는 각 사이트에서 수집된 데이터와 사이트의 Identity가 저장되는 장소이고, 마지막으로 PD는 에이전트의 무결성 및 부인방지에 관련된 proof등이 저장되는 장소이다[2].

에이전트가 홈 사이트인 S_0에서부터 마지막 사이트인 S_N까지 이동하면서 정보를 수집한다고 가정하자. 에이전트는 먼저 S_0에서 다음 사이트인 S_1과 secret로 사용할 random number T를 hash한 값 $T_1(=h(T))$을 구하고 T_1을 S_1의 공개키로 암호화(ET_1)한다. 그 다음 과정은 그림 1에 자세히 묘사되어있다. 그림 1과 같이 에이전트가 모든 사이트를 이동한 후 홈 사이트 S_0로 돌아왔을 때 에이전트의 무결성을 증명하게 된다.

증명과정을 살펴보면 먼저 secret 값 T_1과 AD part의 S_2, D_1의 내용을 구한다. 이 값들로 MIC_1 값을 구할 수 있다. 그리고 PD part의 $DS_1(MIC_1)$과 계산된 MIC_1을 비교한다. 다음으로 $T_2(h(T_1))$와 AD part의 S_3와 D_2의 내용과 앞에서 계산된 MIC_1를 구한다. 앞에서와 마찬가지로 MIC_2 값을 구할 수 있다. 똑같이 PD part의 $DS_2(MIC_2)$와 계산된 MIC_2 값을 비교한다. 이런 계산은 마지막 사이트인 S_N까지 증명을 마치게 되거나, signature 증명이 실패할 때까지 계속된다. 만약 N까지의 증명이 성공했다면 에이전트의 무결성이 증명되는 것이고, 도중에 signature 증명이 실패했다면 그 이후의 데이터에 대해서는 무결성이 보장되지 않는 것이다. 즉 에이전트의 데이터가 변조되었

거나 위조되었음을 알 수 있다.

이러한 MH 프로토콜을 이용하여서 에이전트가 데이터를 수집하고 그 데이터에 대한 무결성을 검증할 수가 있다. 하지만 사용자가 수집된 데이터를 이용해서 어떤 물품을 주문하고자 할 때 사용자의 서명이 필요한데, MH 프로토콜은 이러한 방법에 대해서는 제시하는 바가 없다. 다음절에서는 사용자의 서명을 에이전트가 어떻게 안전하게 행하는지에 대해 살펴보고자 한다.

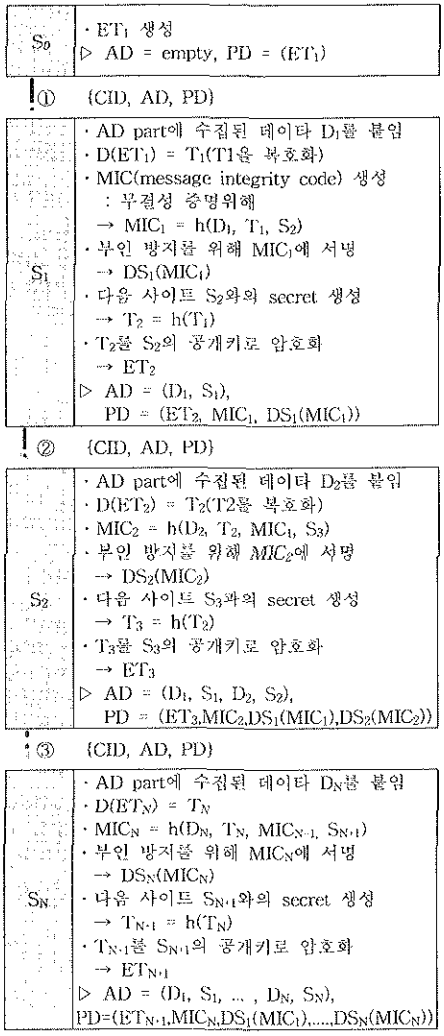


그림 1. MH 프로토콜 진행도

3. 사용자 비밀키의 안전한 대리사용

홈 사이트로 돌아온 에이전트의 무결성이 증명된 이후에는 사용자가 예약 및 주문 등을 위해 에이전트에 의해 수집된 데이터를 비교 분석하여 에이전트를 적당한 사이트로 이동을 시킨다. 여기서 살펴볼 수 있는 것은 사용자와 판매자간에 client/server의 관계가 성립된다는 것을 볼 수 있다. 에이전트가 적당한 사이트로 이동된 후에는 예약 및 주문을 하기 위해 사용자의 서명이 필요하게 된다. 하지만 사용자는 그의 개인키를 노출시키지 않고 주문을 위한 메시지에 서명하기를 원한다. 이를 위해서 Kotzanikolaou 등은 Sander와 Tschudin이 제안한 Undetachable Signature와 CEF(Computing with Encrypted Function)를 이용하여 RSA signature를 기반으로 사용자의 비밀키를 노출시키지 않고 서명할 수 있게 하는 방법을 제안하였다[3].

먼저, 앞으로 사용될 간단한 표기법을 설명하자면, h는 해쉬 함수, Server_ID는 서버의 Identity, Client_ID는 사용자의 Identity, bid_S는 사용자의 주문에 대한 서버의 주문 정보이다.

CEF라는 개념을 살펴본다면, 어떤 electronic shop에서 물품을 구매할 때 사용자의 signature 함수 s가 사용된다. 그래서 signature 함수 s를 보호하기 위해서 사용자는 함수 f와 함께 암호화($f_{signed} := s \circ f$)한다. 그리고 에이전트에게 실행될 수 있는 code인 (f(.), $f_{signed(.)}$)의 함수 쌍, f(.) = $h^{(.)}$ mon n, $f_{signed(.)} = k^{(.)}$ mod n을 주게되고, 서버는 입력값 x(h(Server_ID, Client_ID, bid_S))를 가지고 에이전트가 가지고 있는 함수의 쌍을 실행하게 된다. 그러면 그림 2와 같은 서명 값이 생성된다.

$$f(x) = m, \quad f_{signed(x)} = s(f(x)) = s(m)$$

그림 2. 서명 값

RSA 서명은 그림 3에 간단히 설명하였다.

1. 같은 크기의 큰 random 수 p, q를 생성
2. n = pq, ϕ = (p-1)(q-1)을 계산
3. $1<e<\phi$, $\gcd(e, \phi) = 1$인 random 수 e 선택
4. $1<d<\phi$, $ed \equiv 1 \pmod{\phi}$인 유일한 수 d 계산
5. 개인키는 d 이고, 공개키는 n과 e이다.

그림 3. RSA 서명 기법

$k(=h^d \bmod n)$는 h에 대한 사용자의 RSA 서명이다. f_{signed} 는 사용자의 RSA 함수 $s(.) = (.)^d \bmod n$의 암호화된 함수 $s \circ f$ 이다. 즉 그림 4과 같이 유도된다.

$$f_{signed(.)} = s \circ f(.) = s(f(.)) = s(h^{(.)}) = (h^{(.)})^d = (h^d)^{(.)} = k^{(.)}$$

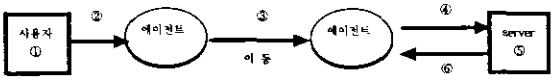
그림 4. 서명 값의 유도식

에이전트가 보내진 후 에이전트의 수행단계에서는 이동 에이전트는 서버에 의해 앞에서 설명한 입력 x 값을 통해 실행된다. 그러면 그림 5와 같이 RSA 서명 값, undetachable signature인 m과 z를 얻게 된다. 따라서 에이전트는 사용자의 비밀키의 노출 없이 서명을 완성하게 된다.

$$m = f(x) = h^x \bmod n$$
$$z = f_{signed(x)} = k^x \bmod n = (h^d)^x \bmod n$$
$$= (h^x)^d \bmod n = m^d \bmod n = s(m)$$

그림 5. RSA 서명값 m, z(undetachable signature)

3절에서 언급된 프로토콜의 개략도는 그림 6과 같다. paramater는 앞에서 설명한 RSA에서 사용된 것과 동일하다. d는 사용자의 개인키, n과 e는 사용자의 공개키, C는 *Client_ID*이다.



① user computes : h = hash(C, req_C) k = h^d mod n f(.) = $h^{(.)}$ mod n $f_{signed(.)}$ = s∘f(.) = $k^{(.)}$ mod n ② C, req_C, f(.), $f_{signed(.)}$	④ C, req_C, f(.), $f_{signed(.)}$ ⑤ server가 함수 실행 f(x) = h^x mod n = m $f_{signed(x)}$ = k^x mod n = m^d mod n = z ⑥ (S, C, bid_S), x, (m,z)

그림 6. RSA기반의 undetachable signature 스킴

본 절에서 언급한 프로토콜을 사용함으로 사용자가 서버에 있는 메시지에 안전하게 서명을 할 수 있게된다. 즉 안전한 구매가 이루어지는 것이다.

4. 제안하는 프로토콜

MH 프로토콜이나, Kotzanikolaou 등이 제안한 프로토콜 모두 그 나름대로의 단점을 가지고 있다. MH 프로토콜은 에이전트가 정보를 수집하고 그 정보에 대한 무결성을 제공한다는 점에서는 장점을 가지나 어떤 사용자가 에이전트가 수집한 정보에 대해서 구매나 예약을 하려고 할 때 필요한 사용자의 서명에 대해서는 언급되지 않고 있다. 또 Kotzanikolaou 등이 제안한 프로토콜은 사용자의 서명에 대해서는 좋은 서명법을 제안하고 있지만 이 프로토콜은 client/server의 관계만 고려하고 있다. 그래서 MH 환경에서의 프로토콜은 적당하지 않다. 또한 부인방지를 위한 서버의 서명도 포함되지 않았다.

제안하는 프로토콜에서는 그 단점들을 해결하기 위해서 두 가지 프로토콜을 하나로 만들고 몇 가지 기법들을 추가하는데 초점을 두었다. 각각의 프로토콜은 2절과 3절에서 분석한 것과 동일하고, Kotzanikolaou 등이 제안한 프로토콜의 ⑥에서 부인 방지를 위해서 x값을 서버의 개인키로 서명을 하였다(x → DS(x)). 두 프로토콜을 합한 상태의 제안하는 에이전트의 경로는 그림 7에 묘사되었다.

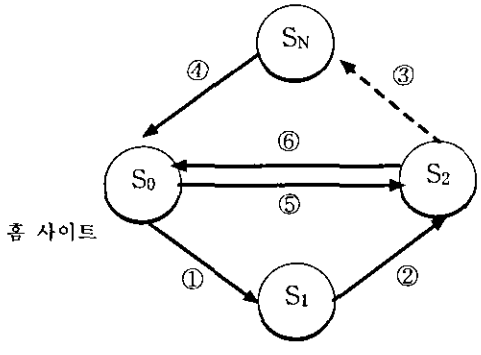


그림 7. 에이전트 이동 개요도

그림 7의 ①~④까지는 에이전트가 데이터의 privacy를 이루면서 데이터를 수집하면서 사이트들을 이동하는 부분이고, 마지막 사이트를 지나 홈 사이트에 도착해서는 에이전트의 무결성의 증명과 데이터의 비교 분석 후 적당한 사이트로 에이전트를 보내게 되는 것이다. ①~④까지는 2절에서 설명한 MH 프로토콜에 해당한다. 이 프로토콜을 사용함으로써 에이전트에 의해서 수집된 데이터에 대한 무결성이 제공되는 것이다. 데이터의 무결성이 증명이 된다면, ⑤에서 에이전트는 적당한 사이트로 이동해서 사용자를 대신해서 예약 및 주문을 하게 되고, ⑥에서 결과를 가지고 돌아오게 되는 것이다. ⑤와 ⑥에서는 3절에서 설명한 Kotzanikolaou 등이 제안한 서명법이 적용되는 것이다. ⑤와 ⑥에서 Kotzanilolaou 등이 제안한 서명법을 사용함으로써 사용자의 비밀키를 노출시키지 않고 서버의 메시지에 안전하게 서명할 수 있는 것이다. 즉 여기서 제안한 방법을 사용함으로써 MH 환경에서 이동 에이전트를 이용한 데이터의 무결성이 보장되고, 안전한 구매가 가능하게 된다.

5. 결론

본 논문에서는 Corradi 등이 제안한 MH 프로토콜과 *Kotzanikolaou* 등이 제안한 서명법을 분석하고 그 단점을 해결하였다. 또한 제안한 두 가지 방법을 접목시키면서 MH 환경에서의 안전한 구매가 일어날 수 있는 방법을 제안하였다. 하지만 앞으로도 이동 에이전트의 근본적인 문제점들의 해결을 위해 더욱더 다양한 연구가 지속적으로 요구된다.

6. 참고문헌

[1] Antonio Corradi, Marco Cremonini, Cesare Stefanelli, "Locality Abstractions and Security Models in a Mobile Agent Environment," IEEE 1998.

[2] Antonio Corradi, Rebecca Montanari, "Mobile Agents Integrity in E-commerce Applications," IEEE 1999.

[3] Panayiotis Kotzanikolaou, Mike Burmester and Vassilios Chrissikopoulos, "Secure Transactions with Mobile Agents in Hostile Environments," LNCS vol.1841, Springer-Verlag, pp. 289-297, 2000.

[4] Tomas Sander, Christian F. Tschudin, "Toward Mobile Cryptography," IEEE, Proceedings of Security & Privacy'1998.

[5] Antonio Corradi, Rebecca Montanari, "Security Issues in Mobile Agent Technology," IEEE 1999.

[6] William M. Farmer, Joshua D. Guttman, and Vipin Swarup, "Security for Mobile Agents: Issues and Requirements," In Proc. of the 19th National Information Systems Security Conf., p 591-597, 1996.

[7] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, "HANDBOOK of APPLIED CRYPTOGRAPHY," pp 425-438.