농산물 모바일 상거래를 위한 효과적인 에이전트 보안 메커니즘

정창렬* · 송진국** · 고진광*

Security Mechanism of Agent for Effective Agro-Foods Mobile Commerce

Chang-ryul Jung* · Jin-kook Song** · Jin-gwang Koh*

이 논문은 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업(IITA-2005-(C1090-0501-0022)의 연구결과로 수행되었음.

요 약

에이전트의 이용은 농산물 모바일 상거래의 유용한 요소이지만, 에이전트 보안이 필수 불가결하다. 만약 공개키 기반 구조를 에이전트 보안에 적용하면, RSA의 공개키 수행 계산과 인증서 실행을 위한 인증기관과의 연결이 필요하기 때문에 에이전트 사이즈가 커지게 되고, 처리속도가 느려지게 되고, 검사 속도가 느려진다. 본 논문에서는 키 분배의 문제점을 개선하여 암호화 키 체인 관계에 의한 호스트 간 상호 인증과 데이터 암호화를 수행하는 메커니즘이다. 이 메커니즘은 에이전트 암호 키 모듈과 난수 생성으로 에이전트의 안전성을 보장받는다. 제안된 메커니즘은 모바일 상거래의 활성화를 위한 안전한 에이전트 보안과 효율성을 고려였으며, 에이전트가 견고하면서 메모리 오버플로우의 문제를 최소화하는 안전한 메커니즘이다.

ABSTRACT

To utilize actively the agent which is one of the elements of revitalization of Agro-Foods Mobile E-commerce, an essential prerequisite is agent security. If using partial PKI(Public Key Infrastructure)-based confirmation mechanism providing security for the agent, the size of agent is becoming larger, the result of the transmission speed is slow, and the confirmation speed is tardy as well because of performing calculation of public keys such as RSA and needing linkage with the CA for the valid examination of certificates. This paper suggests a mechanism that can cross certification and data encryption of each host in the side of improving the problems of key distribution on agent by shaping key chain relationship. This mechanism can guarantee the problem of key distribution by using agent cipher key(ACK) module and generating random number to fit mobile surroundings and to keep the secret of the agent. Suggested mechanism is a thing that takes into consideration security and efficiency to secure agent for the revitalization of M-Commerce, and is a code skill to make the agent solid and is a safe mechanism minimizing the problems of memory overflow.

키워드

Mobile Commerce, Agro-Foods, Agent Security, Integrity

* 순천대학교 컴퓨터과학과

** 진주산업대학교 컴퓨터공학과

접수일자 : 2006. 5. 11

Ⅰ. 서 론

인터넷의 급속한 확산은 의사소통 방식과 개인의 경제적인 활동 영역이 시공간을 초월하여 이루어지고 있다. 이는 경쟁력 확보를 위해 국가 초고속 통신망 구축사업이 추진과 함께 그동안 구조적으로 취약한 농업 경쟁력 강화와 수입 농산물 급증 등으로 인한 대외 경쟁력을 갖추기 위해 농촌 정보화에 많은 노력을 하고 있다[1]. 또한 농촌의 새로운 성장 동력원이자 블루오션으로 대두되고 있는 농촌 어메니티를 개발과 농산물의 전자상거래를 위한 정보 네트워크가 구축되고 정부의 정책적 지원도 이루어지고 있다[2]. 특히 농산물의 시장 주도권을 생산자에서 소비자 중심으로 바뀌고 있으며, 소비구조 또한 고급 농산물 중심으로 변화고 있다. 이러한 변화에 대응하기 위해서는 농촌의 경쟁력과 농산물 시장이 활성화 될 수 있도록 전자상거래가 활성화되어야 한다. 이를 위해서는 사용자 편의를 위한 다양한 응용 기술들이 필요하다. 이때 사용된 기술이 에이전트 기술로 소프트웨어적으로 이동성과 자율성의 특징을 지니고 있어 전자상거래에 많이 응용되고 있다. 그러나 전자상거래에 이용되는 에이전트 기술은 자율적인 이동성으로 인해 악의적인 위협요소들로부터 안전성에 대한 문제가 있다. 특히 전자상거래가 아닌 모바일 상거래에서는 전송할 때 발생하는 트래픽의 고려가 우선시 되어야 한다. 일반적으로 에이전트 보안을 위해 이용되는 PKI는 전송 트래픽에 대한 문제가 있다. 때문에 [4]에서 이용하는 보안 메커니즘은 공개키 기반 구조에서 이루어짐으로 에이전트의 암호화를 할 경우 공개키의 연산 수행에 대한 처리 속도 문제가 발생한다. 또한 인증 수행을 위해서는 인증기관(CA)에 의해서 인증서를 인증 받아야 함으로 인증 속도 문제도 있다. 즉 이들의 문제점은 근본적으로 모바일 상거래에서 수행되기 위해서는 많은 오버헤드가 발생한다. 그렇기 때문에 [5]는 인증 속도 문제를 개선한 MAC(message authentication code)기반을 통해 처리 속도의 문제점을 개선하였다. 하지만 에이전트가 이동하는 동안 기밀성을 유지하기 위해서 비밀키를 공유해야 한다. 이는 키를 분배하기 위한 키 분배 프로토콜이 필요하다. 그렇지 않으면 모바일 상거래에서 에이전트를 사용하는데 있어 신뢰도를 떨어뜨리는 요인이 된다. 또 다른 연구 [6]은 악의적인 에이전트 서버들로부터 에이전트를 보호하기 위해 생성된 일회용 키와 일방향성 해시 함수를 이용하여 키 체인을 통한 데이터의 기밀성과 무결성을 보장한다. 일회성 키를 생성하기 위해서는 일회성 키를 생성하는 서버에 대한 의존도가 높아져 모바일 환경에는 적합하지 않다.

본 논문에서는 이러한 암호화 키 분배의 문제점과 에이전트 송·수신자간의 상호 인증을 통해 에이전트 무결성을 보장한다. 또한 에이전트의 기밀성을 유지하기 위해서 에이전트를 암호화하기 위해 모바일 환경에 적합하도록 에이전트 랜덤 난수에 의한 암호 키를 생성하여 에이전트를 암호화한다. 그리고 이들 암호 키는 에이전트가 이동하는 목적지에서 생성된 랜덤 난수와 해시 함수에 의한 키 체인 값을 XOR하여 키 생성을 새롭게 함으로 모바일 환경에서 더욱 견고하고 안전할 수 있어 에이전트의 상호 인증과 무결성, 그리고 데이터 보호가 이루어지게 된다. 이는 에이전트를 이용한 농산물 모바일 상거래에서 안전한 거래가 이루어질 수 있도록 모바일 환경에 적합한 에이전트 보안 메커니즘을 제시하고 안전성을 분석한다.

Ⅱ. 모바일상거래와 보안

모바일 네트워크를 사용해서 이행되는 모든 가치 전달 활동으로 모바일 상거래는 PC 단말기를 대신하여 거래가 이루어질 수 있도록 지원된다. 개인화된, 경량화된, 지역 정보 제공이 가능한 개인용 hand-held기기들을 이용한 전자적인 거래의 모든 형태들로 고객의 소유권(customer ownership), 개인화(personalization), 위치 기반 서비스(localization), 편재성(ubiquity), 시간 독립성(timeless), 편의성(convenience)등을 제공한다.

에이전트는 모바일 상거래 시스템에서 구매자가 구매욕구가 생기는 시점에서 구매자가 어디에 있든 상관하지 않고 구매자를 대신하여 상품과 판매자를 검색해 주고 구매자의 프로파일과 과거 행동을 참고하여 협상하는 등 일련의 상거래 활동을 지원한다. 그럼으로 구매자가 장시간 구매 행위에 매달리고 무선 연결을 시도함으로써 생기는 비용과 노력을 줄일 수 있다. 하지만 휴대 단말기의 제한적 메모리, 작은 인터페이스, 통신 속도로 인하여 에이전트를 휴대 단말기에서 직접 생성하기 어려운 점과 에이전트에 대한 보안 문제를 고려해야 한다. 이렇듯 모바일 환경에서의 에이전트 시스템은 분산 어플리케이션 환경에 유연하지만 보안에 관련된 심각한 문제가 있다. 이런 문제점들은 악의적인 에이전트나 호스트에 의한 공격들과

외부의 악의적인 요소들에 의한 공격들이다. 그중 에이전트에 의한 호스트의 공격은 호스트의 접근 제어 시스템과 인증을 통해서 어느 정도 해결이 되고 있다[1][7]. 그러나 악의적인 호스트에 의한 에이전트의 공격에 대한 보호는 아직 어려운 문제로 제시되고 있다. 왜냐하면 호스트의 플랫폼은 에이전트의 코드와 상태 등이 호스트에게 완전 노출되기 때문에 에이전트의 코드나 상태가 변경이나 수정될 수 있다. 에이전트의 코드는 생성된 후 변하지 않으므로 생성자의 디지털 서명을 통해서 코드가 변경되었는지에 대한 어느 정도의 확인은 가능하다. 그러나 에이전트의 실행 상태는 항상 변화하기 때문에 악의적인 요소에 의한 공격에 대해 취약하다. 만약 에이전트에 대한 실행 정보가 보호되지 않으면 에이전트의 안전한 운영과 실행이 불가능하다.

그렇기 때문에 이동 에이전트를 보호하기 위한 보안 기법들이 필요하다. 이는 모바일 환경이나 일반 네트워크 환경에서 동일하다. 에이전트를 보호하기 위해서는 에이전트를 공격하는 다양한 방법에 대한 고찰이 필요하나, 본 논문에서는 별도의 공격 방법들에 대해 기술하지 않고 주된 공격자를 악의적인 호스트로 한정한다. 그러나 공격이 유도되는 기본적인 원리의 구조를 에이전트 프로그램이나 공격자 프로그램 로더를 이용하는 RASPS(random access store procedure plus stack)라는 추론 머신을 통해 파악한다[8]. 에이전트가 에이전트 플랫폼에서 초기화하는 동안에 하나의 머신은 에이전트 프로그램에 의해서 로드되고, 다른 하나는 공격자의 프로그램이 공격자 RASPS 안에 로드될 때 에이전트 플랫폼으로 들어와 명령을 해독한다. 그 다음 에이전트의 프로그램 카운터를 계산하고, 프로그램의 실행을 분석, 에이전트의 다음 프로그램 카운터를 자신의 것으로 변경하는 등의 공격을 한다. 에이전트 추론 머신에 의해서 이루어지는 공격의 모델은 [그림 1]과 같다.

에이전트 RASPS에 의해서 공격자는 RASPS의 시스템 코드를 요구하여 에이전트의 속성을 읽고 그 속성을 조작하여 실행을 제어한다. 또한 공격자는 여러 환경들을 스스로 제어하여 에이전트의 명령을 해독하고 실행을 분석하여 공격한다. [그림 2]와 같이 에이전트가 있을 때 에이전트의 공격 형태는 다음과 같다.

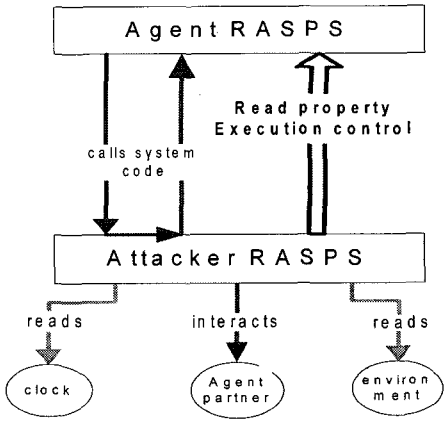

그림 1. 공격 모델
Fig. 1 An attack model

첫째, 구매 에이전트가 호스트에 도착하였을 때 신뢰성 있는 호스트로 위장하여 전달된 에이전트를 가로채기하여 에이전트를 위협하는 위장이 있다. 또한 호스트가 위장을 하여 에이전트에게 악의적인 코드를 삽입하여 정상적인 실행이 이루어지지 못하게 만든다. 둘째, 구매 에이전트가 구매 정보를 수집하기 위해 경쟁사의 호스트를 방문할 때, 경쟁사의 악의적인 호스트가 에이전트의 실행 부분에 포함되어 있는 가격 정보를 찾아내어 가격 정보를 변경한다면 정상적인 가격 정보는 변질된다. 또는 정상적인 최적의 가격 정보를 제공하는 "red rose"를 파는 샵(shop)의 리스트를 변경하거나 누락시킴으로서 에이전트의 정보가 변경된다. 셋째, 호스트에서 에이전트가 임무 수행을 완벽하게 하기 위해 다른 호스트와 협업관계가 요구되기도 한다. 이때 호스트가 에이전트의 통신을 탐색하여 에이전트 통신을 조작할 수 있다. 즉 호스트는 "buy"라는 명령 조작에 의해서 최저가를 제공하는 호스트에 접촉할 것이다. 이 경우 악의적인 호스트는 다른 샵에서 구매하도록 하거나 구매 운영 파라미터에 의해 구매 동작을 조작한다.

넷째, 호스트는 에이전트의 실행을 완벽하게 제어하기 때문에 구매 에이전트의 실행을 지연시키거나 더미(dummy) 코드를 삽입하여 에이전트가 임무 수행을 위한 활동을 멈추게 하거나, 프로그램 명령어의 의미를 변경하여 잘못된 실행을 하여서 그 의미의 해석을 달리 하게 한다. 다섯째, 그 외에도, 호스트에서 만약에 getProvider()와 getAdderess()등의 시스템 콜을 한다면 호스트는 에이전트에게 부정확한 결과들을 리턴 시킨다. 이보다 더 강력

한 공격은 에이전트가 정상적인 실행을 끝내지 않았는데도 강제로 인터럽트를 걸어서 에이전트의 이동 계획과 관계없이 중간에 에이전트 실행을 종료하게 만드는 경우 등이 있다.

```
public void startAgent(){
    if (shoplist == null){
      shoplist = getTrader().getProvidersOf("BuyFlowers");
      go(shoplist[1]);
      break;
    }
    if (shoplist[shoplistindex].askprice(owers) < bestprice){
      bestprice = shoplist[shoplistindex].askprice(owers);
      bestshop = shoplist[shoplistindex];
    }
    if (shoplistindex (shoplist.length - 1)){
      buy(bestshop,owers,wallet);
      go(home);
      if (location.getAddress() == home){
        location.put(wallet);
        }
    }
    go(shoplist[++shoplistindex];
}
```

그림 2. 농산물(장미꽃) 구매 에이전트
Fig. 2 Purchase agent of an Agro-Foods

이러한 에이전트에 대한 공격으로 부터의 보안 문제가 대두 되면서 에이전트를 보호하기 위한 방법들은 에이전트를 보호하기 위해 호스트의 신뢰도를 제3의 신뢰 기관에서 보장하는 방법, 부정 조작을 할 수 없도록 하는 하드웨어를 통한 방법, 그리고 소프트웨어적인 방법 등이 있다. 호스트의 신뢰도를 제3의 신뢰기관에서 보장하는 방법은 모든 호스트의 정보가 한 기관에서 관리되는 점과 알려지지 않는 호스트를 신뢰되지 않은 호스트로 간주하여 에이전트를 보내지 않는 경우가 발생하는 문제점으로 에이전트 보호에 적용하기에 적합하지 않다.

또한, 부정조작을 할 수 없는 하드웨어를 통한 에이전트 보호 방법으로 자바 기반의 스마트카드를 신뢰받는 컴퓨팅 기반으로 사용하여 에이전트가 자료를 안전하게 저장하고 전송할 수 하였다. 그리고 TFE(tamper-proof environment)라는 부정 조작할 수 없는 하드웨어와 CryPO 프로토콜을 사용하여 이동 에이전트의 이동 계획을 보호한다. 그러나 이동 에이전트를 보호하기 위하여 특수하게 제작된 하드웨어를 구입하여야 하고, 추가적으로 하드웨어를 안전하게 배포하는 문제 등의 오버 헤드가 발생하기 때문에 실제 환경에 응용되어 사용되기 어렵다. 그리고 소프트웨어적인 방법은 현재 많이 사용되고 있는 메커니즘적인 방법으로 에이전트의 보호 기법에서 응용되고 있다.

Ⅲ. 모바일 상거래를 위한 에이전트 보안 메커니즘

3.1 신뢰성을 보장을 위한 에이전트 상호 인증

무선 통신 네트워크를 통해 이루어지는 모바일 상거래에서 에이전트와 호스트는 단순히 메시지의 기밀성을 유지하는 것만으로는 충분하지가 않다. 수신자의 입장에서는 수신된 메시지가 전송되어지는 과정에서 불법적인 제3자에 의해서 의도적으로 변조되지 않았다는 확신을 갖게 하는 메시지 무결성과 수신된 메시지에 대한 작성자를 확인하는 인증이 요구된다. 상거래에서 에이전트의 경우도 마찬가지이다. 이는 에이전트가 실행 코드에 의해서 실행되기 때문이다. 에이전트는 이동이 동적으로 결정되며, 자유로운 순회가 이동 계획에 의해 이루어진다. 이때 축적된 에이전트 데이터의 조작이나 자원들의 정보가 추출되는 등의 공격으로부터 무결성 보장하기 위해 해시 체인을 이용한다. 이로써 공격이 발생한 경로나 지점을 찾아낼 수 있어 악의적인 공격으로부터 에이전트를 보호한다.

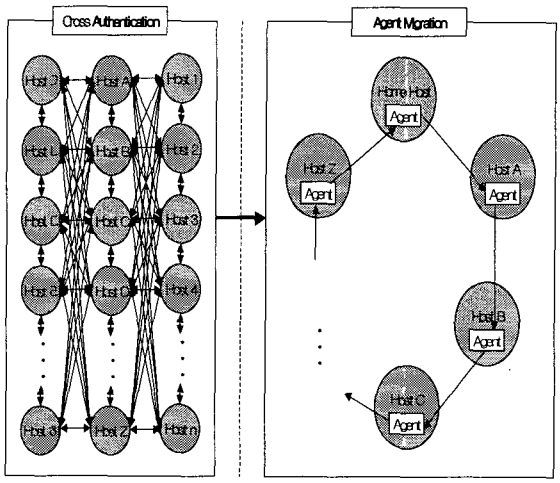

그림 3. 에이전트의 상호 인증과 순회 과정
Fig. 3 Cross certification and circulation of agents

[Step 1] $m_1: H \rightarrow D\{ENC_D^{Pub}(RN_{Home}), Ident_H, Time-stamp, Acc\}$

[Step 2] $m_2: D \rightarrow H\{ENC_H^{Pub}(RN_{Home}, RN_{Det}), Ident_H, Ident_D, Time-stamp, Acc\}$

[Step 3] $m_3: H \rightarrow D\{ENC_D^{Pub}(RN_{Det}), Ident_H, Ident_D, Time-stamp, Acc\}$

[Step 4] $m_4: D \rightarrow H\{Ident_D, H(m_1), M(Accept)\}$

$Ident_H$: $Identifer\ of\ Home\ Host$, $Ident_H$: $Identifer\ of\ Destination\ Host$, $H(x)$: $hash\ Function$

RN_{Home}: $Random\ \nu mber\ of\ Home\ Host$, RN_{Det}: $Random\ \nu mber\ of\ Detination\ Host$,

ENC_H^{Pub}: $Public key\ of\ Home\ Host$, ENC_D^{Pub}: $Public\ Key\ of\ Desination\ Host$

m_i: $Message$, $M(Acc)$: $Accept$ or $Reject\ Contents$, $Time-stamp$: $Time\ stamp$

그림 4. 홈 호스트와 목적지 호스트의 상호 인증 절차
Fig. 4 A cross certification process of home and destination host

뿐만 아니라 혼란된 코드(obfuscated code)를 삽입하여 코드의 실행이나 해석을 어렵게 하는 공격들의 시도를 불가능하게 만들어 에이전트를 보호한다. 즉 DoS(Denial of Service)와 같은 공격은 직접적인 공격 보다 처리 시간을 지연하거나 리소스를 보류하여 정상적인 실행 시간보다 더 긴 시간 동안 실행되게 하는 등 매우 교묘한 형태로 악의적인 호스트에 의해서 이루어지는데, 이런 경우, 타임스탬프를 이용하여 실행 시간을 체크하여 DoS의 공격으로부터 방어한다. 그리고 에이전트가 서로의 신뢰를 보장하고 안전한 실행을 보장하기 위해서 상호 인증 과정을 수행한다. 이는 농산물의 정보에 대한 모바일 상거래에서 홈 호스트가 구매 에이전트를 목적지 호스트에 보낸다고 가정 하자.

홈 호스트는 목적지 호스트에 에이전트를 전송하기 전에, 목적지 호스트가 신뢰할 만한 호스트인가를 검사할 것이다. 또한 목적지 호스트는 전송된 에이전트가 신뢰할 만한 대상(호스트)에서 보내왔는가와 수신된 에이전트가 신뢰할 수 있는가를 검사한다. [그림 4]는 홈 호스트와 목적지 호스트가 서로에 대한 신뢰를 보장하기 위해 검사하는 상호 인증 절차이다. 홈 호스트에서 목적지 호스트로 전송됨을 나타낼 때는 H → D로 표현한다.

[Step 1]은 홈 호스트가 m_1을 송신할 때 생성한 난수(RN_{Home})를 목적지 호스트의 공개키를 이용해서 암호화한 후 암호화한 값과 홈 호스트의 확인자($Ident_H$)를 함께 묶어 목적지 호스트에 전송하는 과정이다. $Time-stamp$는 유일성을 확인하는 확인자와 실행 요청에 포함된 모든 메시지들의 재실행 공격(replay attack)으로부터 보호한다.

[Step 2]는 목적지 호스트가 암호화 된 홈 호스트의 난수(RN_{Home})를 자신의 비밀키로 복호화하여 홈 호스트의 공개키를 이용해서 암호화한 목적지 호스트의 난수(RN_{Det}) 및 홈 호스트와 목적지 호스트의 확인자와 함께 홈 호스트로 전송하는 과정이다. 홈 호스트는 받은 내용을 자신의 비밀키로 복호화 한다. 그 후, 자신이 생성한 난수(RN_{Home})와 목적지 호스트에서 받은 난수(RN_{Home})를 비교해서 두 값이 같다면 홈 호스트는 목적지 호스트를 인증한다.

[Step 3]은 홈 호스트가 암호화된 난수를 자신의 비밀키로 복호화하여 목적지 호스트로 전송한 난수(RN_{Home})와 자신이 가지고 있던 난수(RN_{Home})를 비교한 후, 복호화한 난수(RN_{Det})를 목적지 호스트의 공개키를 이용해서 암호화한 다음 목적지 호스트로부터 받은 난수(RN_{Home})를 자신 및 목적지 호스트의 확인자와 함께 묶어 목적지 호스트로 전송하는 과정이다. 홈 호스트는 복호화해서 얻은 목적지 호스트의 난수(RN_{Det})를 종전과 같은 방법으로 목적지 호스트에 보낸다. 목적지 호스트는 받은 내용을 자신의 비밀키로 복호화해서 난수(RN_{Det})를 얻고, 자신이 생성했던 난수(RN_{Det})와 홈 호스트에서 받은 난수(RN_{Det})를 비교한다.

[Step 4]는 목적지 호스트가 홈 호스트로부터 받은 난수(RN_{Home})와 자신이 가지고 있던 난수(RN_{Home})를 비교하여 두 값이 같다면 홈 호스트를 인증하고 홈 호스트에게 Accept 메시지를 전송하는 과정이다. 이 과정에서 목적지 호스트는 이용 가능한 정보(확인자, 난수, $Time-stamp$)에 기초하여 요청을 수락($Acc=Accept$)하거나 거절($Acc=reject$)할 것인가에 응답을 해야 한다. 여기서 목적지 호스트가 결정의 결과들을 포함하는 메시지를 응답할 때에는 M을 이용한다. 만약 목적지 호스트가 수락 요청을 거절하거나 수행을 거부한다면 M은 거절을 유발시키는 에러 메시지를 포함한다.

3.2 에이전트의 무결성 보장과 데이터 보호 메커니즘

에이전트를 실행하는 호스트는 에이전트가 실행할 수 있는 환경을 제공하므로 악의적으로 에이전트의 데이터나 수행되는 코드를 수정할 수 있다. 이러한 악의적인 요소들로 인해 농산물에 대한 모바일 상거래가 이루어지는 동안 에이전트의 무결성 보장과 수행 결과에 대한 데이터 보호가 이루어져야 한다. 에이전트 데이터가 모바일 상거래에서 사용될 때 사용자의 요청하는 정보가 악의적으로 위·변조되거나 정보 제공 및 송·수신되는 사실을 부인하는 것에 대한 검증을 통해 에이전트의 무결성 보장과 수행 결과 데이터에 대한 보호가 이루어지도록 한다. 이때 에이전트 암호화 키(ACK)를 통한 일회성 키를 생성한다. 생성된 키는 키 체인을 형성하여 에이전트의 무결성이 이루어질 수 있도록 한다. 에이전트는 홈 호스트에서 출발하여 여러 경로를 거쳐 다시 홈 호스트로 되돌아오는 순회 과정을 거친다. 그 과정에서 에이전트는 다양한 공격에 노출되어 무결성을 보장 받기 어렵고, 에이전트의 수행결과 데이터 또한 보호 받기 어려우므로 전자적인 거래가 이루어지는 모바일 상거래에서는 이러한 부분에서 보안이 선행되어야 한다.

그런데 모바일 상거래에서 데이터 보호와 무결성을 보장하기 위해 기존의 공개키 기반의 키 프로토콜이나 PKI 기반의 디지털 서명이나 암호화 알고리즘을 이용할 경우 시스템에 대한 오버헤드가 많이 발생할 뿐만 아니라 통신 트래픽이 많이 발생하게 된다.

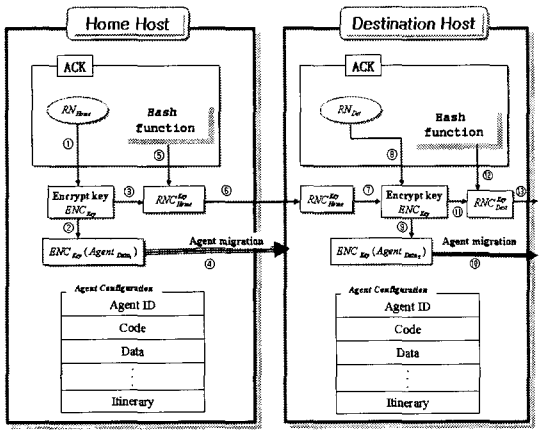

그림 5. 에이전트의 무결성 보장과 데이터보호 메커니즘

Fig. 5 The integrity guarantee and date protection of agents

이러한 측면을 고려하여 간결하고 견고성이 있어야 함은 물론 농산물 상거래를 위한 농산물 정보에 대한 에이전트의 데이터가 기밀성과 무결성을 지녀야 한다. 이를 위해 제시한 메커니즘은 [그림 5]와 같다. 이는 공개키 알고리즘을 이용하지 않고 최소한의 일회성 랜덤 난수를 생성하여 암호화하고 이들의 결성을 위해 일방향(one-way) 해시 함수를 사용하여 체인 관계를 구성한다. 이러한 체인 형성은 홈 호스트와 에이전트의 이동 경로에 의한 목적지 호스트 간에 이루어진다. 에이전트 데이터 암호화와 에이전트 무결성 보장을 위한 메커니즘을 좀 더 상세하게 기술하면 다음과 같다.

① 홈 호스트는 랜덤 난수 RN_{Home}를 생성한다.

② 홈 호스트에서 생성된 난수를 활용하여 에이전트의 데이터를 암호화한다.

③ 암호화에 활용한 난수 ENC_{Key}는 해시 함수에 의한 체인 관계를 형성하기 위해서 이동한다. ④ 에이전트 데이터는 암호화되어 에이전트의 이동 계획에 의해서 목적지 호스트로 이동한다. ⑤ 랜덤 난수로 생성된 암호화 키는 해시함수를 이용하여 해시 체인 값 RNC^{key}_{Home}을 생성한다. ⑥ 생성된 해시 키 RNC^{key}_{Home}는 목적지 호스트와 연결을 하여서 해시 체인을 형성하는 관계를 만든다. ⑦ 홈 호스트와 체인 관계가 형성된 RNC^{key}_{Home}와 목적지 호스트에서 생성된 랜덤 난수 RN_{Det}를 이용하여(⑧), 에이전트 데이터를 암호화할 수 있는 키를 생성한다.

⑨ 생성된 암호화 키를 이용하여 에이전트 데이터를 암호화한다. ⑩ 암호화된 에이전트의 데이터는 에이전트의 이동 계획에 의해 다음 목적지 호스트로 이동한다. ⑪ 암호 키를 다시 해시 함수에 의해서 해시 체인 형성이 이루어질 수 있도록 보낸다. ⑫ ENC_{key}를 해시 함수에 의해서 해시 체인을 형성하기 위해 일방향 해시 체인 값을 생성하여 에이전트의 이동 계획에 의해 다음 목적지로 보내서 해시 체인을 형성한다. 이렇게 에이전트는 해시함수로 암호화 된 키 체인을 이용하여, 에이전트가 이동되기 때문에 이동 과정에서 무결성을 보장 받고, 난수에 의한 일회성 키에 의해 수행 결과 데이터에 대한 기밀성을 보장한다. 에이전트는 실행이 끝나면 그 결과를 에이전트에 안전하게 저장한 후 홈 호스트로 돌아온다. 에이전트가 홈 호스트로 돌아오면 홈 호스트는 무결성과 기밀성을 검사하여 분석을 한다. 만약 신뢰성에 의문이 발생하면 안전하지 않는 것으로 판명하고 해시 체인을 분석하여 문제

가 있는 호스트의 정보 데이터를 분석한다. 이처럼 에이전트에 대한 무결성과 데이터 보호가 이루어지는 메커니즘의 알고리즘은 [그림 6], [그림 7]과 같다.

// 랜덤 난수를 생성
RN_{Home}
/* Home의 랜덤 난수를 이용하여 암호 키 생성*/
$RN_{Home} \rightarrow ENC_{Key}$
// 에이전트 데이터 암호화
$ENC_{Key}(Agent_{Data1})$
// 일 방향성 해시 함수 생성
$RN_{Key_i} = Hash(RN_{Home})$
// 생성된 키 값, 목적지 서버로 전송
$Agent\ Migration(RN_{Key_i})$
// Hash Chain Relation.
$RN_{Key_i} \rightarrow Destination\ Host$

그림 6. 홈 호스트의 에이전트 데이터 암호화와 키 체인 알고리즘
Fig. 6 Data encryption and key chain algorithm of Agents in the home host

이들에 대해 좀 더 구체적으로 설명하기 위해 홈 호스트와 목적지 호스트로 구분하여 기술한다. 홈 호스트는 모바일 환경에서 이용되는 에이전트를 처음 생성하는 생성자의 역할을 한다. 생성된 에이전트가 n개의 호스트를 방문한다고 가정할 때, 홈 호스트는 에이전트의 여정에 포함될 호스트들의 주소 $ip_i(1 \le i \le n)$를 획득한다. 그리고 각각의 호스트에서 n개의 랜덤 난수 RN을 $RN_i(1 \le i \le n)$을 생성한다. 홈 호스트가 생성한 랜덤 난수 RN_{Home}을 가지고 암호 키를 생성하면, 생성된 암호 키를 이용하여 에이전트 데이터를 암호화 $ENC_{Key}(Agent_{Data1})$한다. 그 후 해시 함수를 이용하여 일 방향 해시함수 $Hash(RN_{Home})$를 생성하고 생성된 키 값을 목적지 서버로 전송한다. 전송된 해시 함수 값은 해시 함수에 의한 해시 체인 RN_{Key_i}를 형성한다. 홈 호스트에서 목적지 호스트로 전송이 이루어지면 목적지 호스트는 홈 호스트에서 전송한 에이전트를 수신한다.

// 에이전트 수신
// 생성된 해시 값 수신
RN_{Key_i}
// 랜덤 난수 생성
RN_{Det}
/* 데이터를 암호화하기 위한 암호화 키 생성*/
$ENC_{Key} = RN_{Key_i} \oplus RN_{Det}$
// 에이전트 데이터 암호화
$ENC_{Key}[Agent\ data]$
// 일 방향성 해시 체인 값 생성
$RN_{Key_{i+1}} = Hash(ENC_{Key})$
// 생성된 키 값, 목적지 서버로 전송
$Agent\ Migration(RN_{Key_i})$
// 해시 체인 관계 형성.
$RN_{Key_{i+1}} \rightarrow$ *Next Host*

그림 7. 목적지 호스트의 에이전트 데이터 암호화와 키 체인 알고리즘
Fig. 7 Data encryption and key chain algorithm of Agents in the destination host

수신한 에이전트를 암호화하기 위해 암호 키(ENC_{Key})는 홈 호스트의 해시함수에 의해 생성된 해시 값 RN_{Key_i}을 목적지 호스트가 생성한 랜덤 난수 RN_{Det}를 이용하여 생성한다. 생성된 키로 에이전트 데이터를 암호화한다. 암호화된 에이전트 데이터는 에이전트의 이동 계획에 의해 다음 목적지 호스트로 이동한다. 한편 암호 키 ENC_{Key}는 해시함수에 의한 일 방향 해시 체인 값 $Hash(ENC_{Key})$을 생성하고, 생성된 키 값을 다음 목적지로 전송한다. 전송된 해시 함수 값은 해시 함수에 의한 해시 체인 $RN_{Key_{i+1}}$를 형성하여 에이전트의 무결성을 보장한다.

Ⅳ. 에이전트 수행 결과와 안전성 분석

생성한 에이전트가 이동할 때에는 에이전트 데이터를 안전하게 보호하는 알고리즘에 의해 이루어진다. 수행 과정은 랜덤 난수를 생성하여 에이전트 데이터를 암호화하는 과정과 무결성을 보장하기 위해 [그림 8]과 같이 해시 체인 알고리즘에 의해 이루어진다. 홈 호스트에서 에이전트 데이터는 랜덤 난수를 생성한 후 에이전트의 데이터를 읽고서 이루어진다. 암호화를 한 후에 메시지의 코드 값

을 생성한다. 그리고 에이전트가 서버에 접속되어 있는 홈 호스트의 주소와 연결을 시도하여 신뢰성이 확인되면 서로 연결된다. 이때 에이전트는 정상적으로 임무를 수행한다. 또한 에이전트가 이전 호스트의 이름과 이동하는 호스트의 이름을 제공함으로 이동 되어온 경로를 쉽게 알 수 있다. 송신자의 호스트와 수신자의 호스트의 수행되는 과정을 [그림 8]을 통해서 에이전트가 수행되는 것을 전체적인 흐름을 알 수 있다.

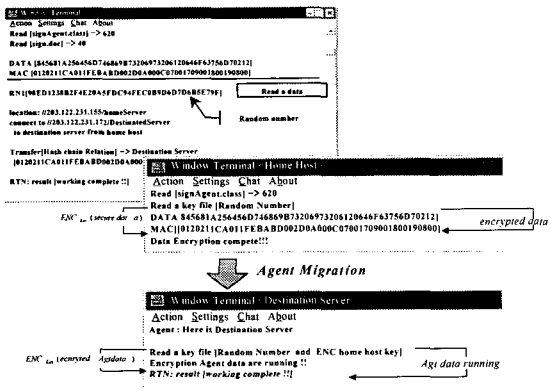

그림 8. 에이전트 수행 결과(무결성 보장과 데이터 보호)
Fig. 8 A execution result of agents(integrity guarantee and data protection)

에이전트가 안전하게 수행되었음을 확인하기 위해서는 두 가지 측면에서 안전해야 한다. 하나는 해시 체인에 대한 무결성의 보장이고, 다른 하나는 에이전트가 송·수신자 간의 상호 인증을 통해서 데이터의 암호화가 이루어질 때 서로에 대해 신뢰가 이루어져야 한다. 뿐만 아니라 암호 키가 서로 교환될 때 위조되지 않았음을 밝히는 것이다. 먼저 해시 함수를 이용한 해시 체인에 대한 안정성을 분석한다.

첫째, 공격자들이 해시 체인을 끊고 자신의 해시 함수를 이용하여 해시 체인을 새롭게 만드는 공격과 해시 체인을 위장하는 공격으로부터 안전하다. 이러한 공격은 네트워크상에서 빈번하게 발생되는 악의적인 요소들의 공격 방법이다. 이 경우 본 논문에서 제안한 메커니즘과 알고리즘을 통해 알 수 있듯이 일방향 해시 함수를 이용한 송신자와 수신자간의 상호 인증이 이루어졌을 때 송신과 수신이 이루어진다. 만약 이를 어기고 새로운 체인을 생성하여 삽입하고자 할 경우, 공격자는 송신자로부터 전송되어 온 RNC_{Home}^{key} 값을 알아야 한다. 그렇기 위해서는 공격자는 송신자와 공모를 해서 앞서 형성한 체인 값 (RN_{Key_i})과 RN_{Det}을 알아야 한다. 그러나 이들의 공격은 현실적으로 어려움이 있어 불가능하다. 그러므로 에이전트에 대한 무결성을 보장할 수 있다

둘째, 암호 키를 송·수신 과정에서 위조 될 경우이다. 이는 공격자가 어느 하나의 체인을 끊고 자신의 체인으로 계속해서 연결하는 알고리즘을 생성하는 경우이다. 이는 공격자 해시 체인을 임으로 삽입하는데 성공하였다고 가정 하에서 이루어지는 공격이다. 이러한 유형은 공격자가 에이전트 데이터를 위조하여 악의적으로 이용하고 할 때 발생하는 공격으로 정상적으로 해시 체인을 끊고, 공격자의 해시 체인을 형성하면 정상적인 해시 체인으로 위장하는 공격이다. 이는 에이전트가 홈 호스트에서 출발하여 연결하고자 하는 목적지 호스트로 이동되기 전에 공격이 이루어졌을 때 가능하다. 그리고 마지막 단계인 홈 호스트로 되돌아오기 이전에 원래의 해시 체인으로 되돌려 놓아야 한다. 하지만, 이런 경우는 이론적으로 가능하나 실제 이루어 질 수 없다. 왜냐하면, 호스트와 호스트 간에 상호 인증을 통해 서로를 신뢰하였을 때 에이전트가 이동할 수 있고, 에이전트가 이동되면 해시 체인 형성으로 에이전트의 데이터를 보호할 수 있도록 암호화키를 생성하여 암호화하기 때문이다. 즉 각 호스트는 수신된 해시 체인RN_{Key_i}을 알았다고 하더라도 RN_{Det}에 의해 생성되는 에이전트 데이터의 암호 키(ENC_{Key})를 알아야 한다. 암호 키는 해시 체인과 랜덤 키에 의해서 생성됨으로 이것이 정상적으로 이루어지지 않으면 어렵다. 더욱이 향후 체인 형성을 하는 데 있어서도 $RN_{Key_{i+1}} = Hash(ENC_{Key})$처럼 형성이 되어야 한다.

그러므로 이러한 유형의 공격은 본 논문에서 제안한 메커니즘에서는 이루어질 수 없다. 그리고 홈 호스트에 에이전트가 수행 결과를 가지고 되돌아오면 홈 호스트에 의해서 해시 함수를 이용하여 다시 무결성을 검증 받기 때문에 어렵다. 또한 에이전트에 대한 공격자들의 가로채기, 도청, 위장 공격 등에 대해서도 안전하게 보호가 이루어진다.

V. 결 론

모바일 상거래의 활성화와 그 활용에서 있어서 중요시되고 있는 에이전트가 효율적이면서 신뢰성이 보장되기 위해서 안전한 에이전트 보호가 필 수 적이다. 이에 본 논

문에서는 농산물의 모바일 상거래가 효율적으로 이루어지기 위해 모바일 환경에 적합한 에이전트 시스템의 자원 보호 및 에이전트의 수행 결과 보호를 위한 보안 메커니즘을 제안하였다. 특히, 악의적인 호스트나 에이전트의 공격으로부터 에이전트를 보호하기 위한 에이전트 데이터 보호와 무결성 보장 메커니즘과 알고리즘을 설계했다. 제안한 메커니즘은 에이전트 데이터 보호와 무결성 보장을 위해 일회성 난수를 생성하여 에이전트 데이터를 암호화 할 수 있도록 암호 키로 설계하였다. 때문에 에이전트 수행 결과인 데이터를 안전하게 암호화하고 무결성 또한 보장할 수 있다. 그리고 제안한 기법은 모바일 환경에서 발생할 수 있는 네트워크의 트래픽을 고려하여 설계하였으므로 메모리의 오버플로우가 발생하는 문제점을 최소화 하였다. 뿐만 아니라 전자상거래의 에이전트 보안에서 많이 쓰이고 있는 PKI 환경의 공개키 기반의 암호화 알고리즘에 비해 속도가 빠르고, 속도를 고려한 MAC 기반의 무결성 메커니즘이 갖는 키 분배 문제에 의한 신뢰도 저하를 개선하였다. 그로인해 제안된 메커니즘은 농산물 모바일 상거래를 위한 효율성과 무결성이 보장된 안전한 에이전트 수행 구조를 지니고 있다.

참고문헌

[1] 하영수, "농촌 정보화 정책 결정에 관한 연구", 대한정치학회보, 제11집 제3호, pp.75-93, 2004.

[2] 민승규, "농정방향과 어메니티 개념도입의 필요성", DDA대응 농촌 경제 활성화와 어메니티 자원 개발 심포지움, pp.5-24, 2003.

[3] G. Karjoth, N. Asokan, and C. Gulcu, "Protecting the Computation Results of Free-Roaming Agents", K. Rothermel and F. Hohl(Eds.) in Proceeding of MA'98 Mobile Agents, LNCS 1477, pp.195-207, Springer-verlag, 1998.

[4] N. M. Karnik, "Security in Mobile Agent Systems", The Graduate School, University of Minnesota, Ph. D. thesis, October, 1998.

[5] T. Taka, T. Mizuno, T. Watanabe, "A Model of Mobile Agent Services" in Enhanced for the International Conference on Parallel and Distributed Systems, pp.274-281, 1998.

[6] J. Y. Park, D. I. Lee, et al. "One-Time Generation System for Agent Data Protection in Mobile Agent Systems,", in Journal of The Korea Information Science Society, Vol. 28, No. 3, pp.309-320, 2001.

[7] N. Karnik and A. Tripathi, "Security in the Ajanta Mobile Agent System", Software- Practice and Experience, 2000.

[8] F. Hohl, "A Model of Attacks of Malicious Hosts Against Mobile Agents," in Proceeding of the ECCOP Workshop on Distributed Object Security and 4th Workshop on Mobile Object System : Secure Internet Mobile Computations, pp.105-120, 1998.

저자소개

정 창 렬(Chang-Ryul Jung)

1999년 순천대학교 대학원 컴퓨터교육 (석사)
2005년 순천대학교 대학원 컴퓨터과학과 (박사)

2003년 6월 : University of Alberta, Canada, Visiting Researcher.
※관심분야 : Information Security, Mobile Agent, Image processing, E-Commerce

송 진 국(Jin-Kook Song)

1990년 홍익대학교 대학원 전자계산학과 (석사)
1998년 홍익대학교 대학원 전자계산학과 (박사)

1998년 ~ 현재 진주산업대학교 컴퓨터공학부 교수
※관심분야 : 프로그래밍언어론, 역컴파일러

고 진 광(Jinl-Gwang Koh)

1982년 홍익대학교 전자계산학과 (학사)
1984년 홍익대학교 대학원 전자계산학과 (석사)

1997년 홍익대학교 대학원 전자계산학과 (박사)
1997년 ~ 1998년 Oregon state University. 컴퓨터공학과 방문 교수
2001년 3월 ~ 2002년 8월 순천대학교 정보전산원 원장
2005년 3월 ~ 현재 순천대학교 공과대학 학장.
※ 관심분야 : 데이터베이스, 전자상거래와 보안