지불프로토콜 보안 검증을 위한 구조
(A Framework for Verifying Payment Protocol Security)

한 국희 (Han, Kook Hee), 권 영직 (Kwon, Young Jig)
(khhan@taegu-c.ac.kr) (yjkwon@daegu.ac.kr)
대구대학교 대학원 컴퓨터정보공학과

요 약

보안의 형식검증은 시스템의 초기 상태를 정의하고 트랜잭션을 통해 그 상태를 추적하며 보안을 위해 각 상태를 체크하는 과정이다. 보안 증명이 잘못될 수 있는 경우는 시스템의 초기상태를 정의할 때와 상태가 안전하기 위한 조건을 정의할 때인데, 본 논문에서는 상태 트랜잭션을 위해 BAN 논리를 이용하여 대표적인 지불프로토콜인 NetBill 프로토콜에 대한 형식기법을 제안하였으며, 보안 프로토콜의 증명을 위해 BAN 논리의 확장을 제시했다. 이러한 확장된 연구결과는 원자성, 익명성 및 프라이버시와 같은 다른 중요한 보안의 특성을 증명하기 위해 이용될 수 있다.

1. 서론

프로토콜이란 어떤 목적을 달성하기 위한 통신 주체간의 일련의 규약을 의미하며, 이와 같은 관점에서 지불 프로토콜이란 구체적으로 암호기술을 이용하여 다양한 전자지불의 목적을 달성하기 위한 규약을 의미한다.

어떤 시스템이 안전한지 확신하는 것이 불가능할지 모르는 반면, 가능한 한 시스템이 타협할 수 없다는 것이 보장되는 일은 중요하다. 그러한 보장은 알려진 공격을 테스트하는 표준 순응도(standards compliance)와 형식 검증으로부터 나올지 모른다.

시스템의 보안 수준을 명시하는 다양한 보안 표준이 존재하는데, 가장 일반적인 것 중의 하나가 National Institute of Standards and Technology(NIST)에 의해 부여된 연방 정보 처리 표준(FIPS; Federal Information Processing Standards)이다. 다양한 FIPS 간행물이 있는데, 각각은 고유의 목적을 지닌다. 그들 중 많은 것들은 보안 및 암호표기법과 관련이 있는데, 가장 관련이 깊은 것은 FIPS 140-1이다[6]. 이는 암호 모듈의 보안 요건을 명시하고, FIPS 800-2는 공개 키 암호법[8], 800-9는 전자상거래를 위한 표준을 명시한다[7].

본 논문에서는 지불프로토콜의 보안검증을 위하여 암호 프로토콜 분석을 위한 최초의 논리적인 도구인 BAN 논리의 확장을 제시하였다. 하드웨어 플랫폼은 명시하지 않았으며, 소프트웨어 보안을 위한 형식 모형을 가지며, 소프트웨어에 대해 레벨 4의 순응도를 지닌다. 또한 이 논문에서 이용된 모든 암호 알고리즘은 FIPS에서 승인된 것이다.

2장에서는 보안의 검토와 형식검증에 대하여 살펴보고 3장에서는 지불 프로토콜 보안 검증을 위한 접근법으로 BAN논리에 대하여 표기법 및 정의와 보안분석에 대하여 살펴본다. 제4장에서는 보안의 형식증명으로 구조를 설명하고 5장에서 결론을 맺는다.

2. 보안 검토와 형식검증

보안을 체크하는 가장 일반적인 방법은 설계 단계 및 구현 이후 수행되는 보안 검토이다. 설계 단계에서는, 많은 연구자들에 의해 검토, 개선되었다.

형식 검증은 프로토콜 고장을 체크하기 위한 가장 잘 알려진 방법 중 하나이다. 형식 검증의 첫 번

째 단계는 시스템의 형식 모델링이다. 그러한 형식 모델은 FIPS 140-1에 따라 레벨 4 소프트웨어 보안에 요구되는 것이다.

암호 프로토콜은, 부적절하게 설계된다면, 메시지 수정 공격 또는 때로는 심지어 도청 공격에 취약하게 만드는 결함을 지닐지 모른다. 형식 검증 접근법은 프로토콜 결함을 탐지하기 위해 제안되었다. 프로토콜 결함은 메시지 교환 및 내용의 논리에 의존하며, 계산 복잡성의 맥락에서 그 강도가 판단되는 암호방식 및 유사 메커니즘의 취약성과는 구별되어야 한다.

안전한 프로토콜 설계를 위해서는 분석 또는 통합 접근법을 취할 수 있는데, 분석 접근법은 원래 제안된 프로토콜 설계에 제약을 가하지 않지만, 설계를 분석하기 위해 형식 기법을 이용하고, 그것이 정확한지를 증명하거나 프로토콜 결함을 탐색한다. 통합 접근법은 설계 원칙 및 테크닉의 식별에 초점을 맞추는데[11], 만약 이를 따른다면 결과로서 생기는 프로토콜의 보안성을 선험적으로 보증한다.

프로토콜을 설계하는 동안 이러한 가이드라인 중 일부를 염두에 두는 것이 중요한 반면, 설계 원칙을 엄격하게 준수하거나 그에 의존하는 것은 적절치 못한 것으로 판명될지 모른다. 그 이유는 다음과 같다.

· 설계 접근법에 의해 예견되지 않은 보안 목표.

· 어떤 예견되지 않은 보안 목표와 명시된 설계 제약의 불일치.

· 프로토콜의 비효율성으로 이어지는 설계 제약의 경직성.

· 의도되지 않은 보안 침해의 결과를 낳는 불충분하게 정확한 설계 제약.

본 논문에서는 이 모든 것을 고려한 후 프로토콜의 보안 및 기타 특성을 형식적으로 검증하도록 했다.

3. 지불 프로토콜 보안 검증을 위한 접근법

형식 검증은 시스템의 초기 상태를 정의하고, 트랜잭션을 통해 그 상태를 추적하며, 보안을 위해 각 상태를 체크하는 과정이다. 보안 증명이 잘못될 수 있는 두 가지 경우는 다음과 같다.

· 시스템의 초기 상태를 정의할 때.

· 상태가 안전하기 위한 조건을 정의할 때.

상태 트랜잭션을 위해 이용하는 논리는 잘 알려진 BAN 논리의 확장이며 아래에서 자세히 설명된다.

3.1 BAN 논리

암호 프로토콜의 정확성을 분석하기 위한 형식 기법의 이용은 1989년 BAN 논리[12]의 개발과 함께 널리 퍼졌다. 그때 이후로 BAN 논리의 문제점을 보고하는 몇 편의 논문 [3, 4, 15]과 이러한 한계 중 많은 부분을 극복하는 BAN-스타일 논리를 제안하는 몇몇 다른 논문들이 발표되었다. 본 논문에서 이용하는 논리는 BAN-스타일 논리[12]의 의미론[1]에 기초를 두고 있다. 이 장에서는 [1]에서 제안되는 BAN 논리를 설명하고, 강력한 침입자를 모델링하기 위해 BAN 논리의 의미론을 확장한다.

3.1.1 언어

principal은 트랜잭션에 관련되는 당사자들, **키**는 이용되는 암호 키, **메시지**는 당사자들 사이에서 교환되는 메시지, **공식**은 다른 소트 그리고 몇몇 논리 및 워드 연산자로부터 구축되는 표현들을 말한다. 언어에서는 전통적인 논리 연산자: $\wedge$, $\vee$, $\rightarrow$, $\forall$와 일치 연산자(identity operator) =를 가지며, 다음의 워드 연산자를 지닌다.

워드 연산자는 전통적인 논리 연산자보다 높은 우선순위를 가진다.

· (*,*) : Message x Message → Formula
· once_said : Principal x Message → Formula
· sees : Principal x Message → Formula
· possesses : Principal x Key → Formula
· shared_key_of : Key x Principal x Principal → Formula
· public_key_of : Key x Principal → Formula
· private_key_of : Key x Principal → Formula
· $[*]^*$: Message x Key → Formula
· rightly_believes : Principal x Formula → Formula

다음의 <표 1>에서는 BAN 논리를 암호화하는데 사용되는 몇 가지 함수들을 보여주고 있다. inv는 K와 K^{-1}의 연관 관계의 의미를 명확하게 해주는 것이고, distinct는 두 명의 다른 당사자를 나타낸다.

3.1.2 모형

환경은 principals의 유한한 집합으로 구성된다. principal P의 로컬 상태는 튜플(B_p, O_p, S_p, K_p)로 정의되며, 다음의 직관적 해석을 지닌다:

B_p : P가 현재 신뢰하는 진술들의 집합.
O_p : P가 전송한 (하위) 메시지들의 집합.
S_p : P가 전송한적이 있는 메시지들의 집합
K_p : P가 소유한 키들의 집합

<표 2> BAN 함수

함 수	설 명
believes(P , X)	*P* **believes** *X*
sees(P , X)	*P* **sees** *X*
said(P , X)	*P* **said** *X*
controls(P , X)	*P* **control** *X*
fresh(X)	**fresh**(*X*)
shared_key(K, P, Q)	$P \longleftrightarrow Q$ (*P와 Q는 서로 통신하기 위해서 암호화 키 k를 공유한다.*)
public_key(K , P)	$\overset{K}{\longmapsto} P$
secret(Y , P , Q)	$P \underset{<}{\overset{Y}{-\!\!>}} Q$
encrypt(X , K , P)	$\{X\}_K$ *from* *P*
combine(X , Y)	$\langle X \rangle_Y$
comma(X , Y)	*X* , *Y* (conjunction)
inv(K , K^{-1})	*K* and K^{-1} are a public/private key pair
distinct(P , Q)	principals *P* and *Q* are not the same

정의 1 *전역 상태(global state)란 환경 속의 각 principal에 대해 principals로부터 로컬 상태로의 사상이다. 앞으로, 한정어 없는 용어 "상태"는 전역 상태를 의미한다.*

논리 및 그 의미론의 공리화(axiomatization)는 [1]에서 발견될 수 있다. 그것은 논리가 주어진 의미론 [1]에 대해 안전하다는 것을 증명했다.

다음에서는, 몇몇 용어가 정의되고, 이것은 후에 보안을 위한 전자상거래 프로토콜을 분석하기 위해 이용되는 정리에서 이용된다.

이 장에서는 시스템 상태 및 프로토콜 작용의 특성과 그 관계를 표현하기 위해 형식적으로 정의된다. 이러한 정의는 프로토콜 분석에 유용한 정리의 개발 및 표현에 이용된다.

일반적인 프로토콜 단계는 한 당사자 (P)에 의해 또 다른 당사자 (Q)로 보내지고 있는 메시지 (X)로 구성되며, P → Q : X로 표현된다.

정의 2 *행동(action) a의 논리적 번역. T(a)는 다음과 같이 정의된다.*

T(P → Q : X)
:= (P once_said X ∧ Q sees X ∧ I sees X)
(I = Intruder)

$T(\varnothing) := \varnothing$

$T(a_1 ; a_2) := T(a_1) \cup T(a_2)$

직관적으로, *T(a)*는 *a*의 실행에 있어 한 상태에서 새로운 상태로의 전이를 계산하는 데 필요한 술어

(predicates)의 최소 집합이다. 새로운 상태는 술어의 집합에 의해 주어진다.

$\{X : \{A \cup T(a) \vdash X\}$

정의 3 *술어 A의 컬렉션과 프로토콜 단계에 대해*

a = P → Q : X, 술어 "A가 a를 허용한다(A allows a)"는 메시지 X의 구조에 대해 귀납적으로 정의된다.

A allows P →Q: X^k , k ∈ K_p
:= for some S: A allows P → S: X
A allows P →Q: X^k , k ∉ K_p
:= A ⊢ P sees X^k
A allows P →Q: (X,Y)

:= A allows P → Q : X
:= A allows
P → Q: X
A allows P → Q : ∅
:= A⊢ P believes ∅
A allows P → Q : X
:= True (all other cases)

직관적으로, *A allows a*는 행동 a가 A가 유지하는 어떤 상태에서의 유효한 행동임을 의미한다. 허용된 행동은, 보내진 메시지가 보여진 메시지 구성요소, 알려진 키, 평문(plain text)의 어떤 결합으로부터 구축되는 것이다.

정의 4 *양의[positive] 공식은 최소한 다음과 같은 집합이다.*

∅	is positive if ⊢ ∅;
k private_key_of P	is positive;
k public_key_of P	is positive;
P believes ∅	is positive if ∅ is positive;
P sees X	is positive;
P once_said X	is positive;
∅∧ Ψ	is positive if ∅ is positive and Ψ is positive;
∅∨ Ψ	is positive if ∅ is positive or Ψ is positive;
(∀x:∅)	is positive if ∅ [x ← u] is positive for all; terms u of the appropriate kind not containing unbound variables.
A finate set of formulas F	is positive if the logical AND of the formulas in F is positive.

직관적으로, *양의 공식*은 한 상태에서 그것이 *참*이라면 그 상태에서 어떤 허용된 행동의 실행 이후에도 *참*으로 남아있는 공식이다. principal이 보거나 일단 말한 공식은 항상 양(positive)이다.

정의 5 *프로토콜 규격은 공식의 두 집합, A(가정) 및 C(결론)에 의해 주어진다. 프로토콜 P는 "A가 처음에 유효할 때" "C가 마지막에 -즉, P를 실행한 이후- 유효하다"는 것을 보증하는 그러한 규격 (표기법: {A}P{C})을 만족시킨다.*

정의 6 *수정 연산[rectify operation] R[*]은 공식을 공식에 대응시킨다. 특히, 그것은 형태 "P believes ∅"의 공식을 "P rightly_believe ∅"에 대응시킨다. 그것은 다음과 같이 정의된다.*

R[P believes Q]	:= P rightly_believes ∅
R[∅ ∧ Ψ]	:= R[∅]∧ R[Ψ]
R[∅ ∨ Ψ]	:= R[∅]∨ R[Ψ]
R[∅ → Ψ]	:= R[∅]→ R[Ψ]
R[$\forall_x$: ∅]	:= ($\forall_x$:R[∅])
R[∅]	:= ∅, other cases

수정 연산은 아래의 정리 1과 같은 진술을 표현하는 데 있어 중요하다.

정리 1 *if A ∪ T(a) is positive, A allows a, A ∪ T(a) ⊢ C, then {R[A]} a{R[C]}.*

3.1.3 표기법

본 논문 전체에 걸쳐 다음의 표기법을 이용한다.

- A는 고객 Alice를 나타낸다.
- a는 Alice의 공개 키를 나타낸다.
- $\frac{1}{a}$ 는 Alice의 개인 키를 나타낸다.
- B는 상인 Bob을 나타낸다.
- b는 Bob의 공개 키를 나타낸다.
- $\frac{1}{b}$ 는 Bob의 개인 키를 나타낸다.
- I는 침입자를 나타낸다.
- 공개 키 i는 침입자의 공개 키를 나타낸다.
- $\frac{1}{i}$ 는 침입자의 개인 키를 나타낸다.
- e와 E는 임의로 생성된 암호화 키를 지칭하기 위해 이용된다.
- $\frac{1}{e}$ 와 $\frac{1}{E}$ 는 상응하는 해독 키이다.

· $P \rightarrow Q$: $[message]^{\perp}_{b}$ 는 P가 P의 개인 키로 서명된 "메시지"를 Q에게 보낸다는 것을 나타낸다.

· $P \rightarrow Q$: $[message]^{p}$는 P가 Q의 개인 키로 보증된 "메시지"를 Q에게 보낸다는 것을 나타낸다.

형태의 표현은 다음과 같다:

$C = \wedge \varnothing_1$
$\wedge \varnothing_2$ refers to $\varnothing_1 \wedge \varnothing_2 \wedge \varnothing_3$
$\wedge \varnothing_3$

3.2 보안 분석

여기에서는 앞절로부터의 논리를 구축하고, 전자상거래 프로토콜의 보안이 어떻게 분석될 수 있는지를 살펴본다.

프로토콜이 안전함을 증명하는 과정은 다음과 같다.

· 시스템의 모델링
· 의문시되는 애플리케이션에 대한 보안의 정의
· 강력한 침입자의 모델링
· 시스템의 초기 상태 정의
· 시스템이 프로토콜의 과정 속에서 변화함에 다른 시스템 분석

각 단계를 설명하면 다음과 같다.

3.2.1 전자상거래 시스템 모형과 보안

(1) 모형

일반적으로, 전자상거래 트랜잭션 프로토콜은 고객(customer)과 상인(merchant) 두 가지 종류의 당사자와 관련된다. 고객은 상품 P와 교환하여 지불할 준비가 되어 있는 일정량의 화폐를 지녀야 한다. 상인은 화폐/지불과 교환하여 기꺼이 팔려고 하는 상품 P를 가져야 한다. 고객이 상인으로부터 상품 P를 받고 상인이 고객으로부터 정확한 지불을 받을 때 트랜잭션(거래)가 완료된다. 이러한 당사자들은 물리적으로 어디에든 위치할 수 있다. 각 당사자에게는 네트워크에 연결된 컴퓨터가 있다. 당사자들은 네트워크를 통해 서로에게 메시지를 보냄으로써 통신한다. 프로토콜은 어떤 순서로 당사자들 간에 교환된 수많은 메시지와 관련되는데, 적절한 트랜잭션 완료라는 결과를 낳는다.

(2) 보안

프로토콜 보안의 증명에 있어 가장 어려운 이슈 중 하나는 보안의 정의이다. 모든 이들은 보안의 침해가 시스템에 일어날 수 있는 "나쁜" 어떤 것이라는 데 동의한다. 그러나, 모든 가능한 "나쁜" 일들을 열거하는 것은 아주 어렵다. 일련의 알려진 보안 공격 하에서 안전하다고 선언된 시스템들이 종종 지금까지 알려지지 않은/예견되지 않은 공격 하에서는 안전하지 않은 것으로 밝혀진다.

그러나, 형식적으로 정의되거나 진술되지 않은 것을 증명할 수는 없다. 본 논문에서는 보안을 가능한 한 폭넓게 정의한다. 이러한 폭넓은 정의는 보다 새롭고 보다 미묘한 보안 공격이 고안될 때에도 유효할 것으로 기대된다. 전자 상거래 시스템을 위해, 보안을 다음과 같이 정의한다.

· 어떤 침입자도 트랜잭션(처리) 도중 또는 이후 어느 때라도 고객 A의 개인 키 $\frac{1}{a}$ 를 지닐 수 없다.

· 어떤 침입자도 트랜잭션(처리) 도중 또는 이후 어느 때라도 상인 B의 개인 키 $\frac{1}{b}$ 를 지닐 수 없다.

· 어떤 침입자도 트랜잭션(처리) 도중 또는 이후 어느 때라도 상품에 대해 이루어지는 저불의 어떤 부분도 지닐 수 없다.

· 어떤 침입자도 트랜잭션(처리) 도중 또는 이후 어느 때라도 처리되고 있는 상품을 지닐 수 없다.

형식적으로, I가 침입자를 지칭한다면, 보안 C를 다음과 같이 정의한다.

$C = \wedge \frac{1}{a} \notin S_I$
$\wedge \frac{1}{b} \notin S_I$
$\wedge\ payment \notin S_I$
$\wedge\ product \notin S_I$

3.2.2 침입자 모형

어떤 보안 침해도 없다는 것을 보여주기 위해서는, 침입자가 얼마나 강력한지 그리고 어떤 종류의 침입자 행동이 가능한지를 아는 것이 중요하다.

침입자는 형식 논리를 이용해 모델링된 적이 없는데 그 이유는 과거에는, 형식 검증이 보안 프로토콜의 정확성 (예, 인증, 키 분배 등)을 증명하는 데 이용되었다. 이러한 프로토콜 정확성의 증명은 보안의 증명과 마찬가지이다. 단지 최근에는, 형식 증명이 프로토콜의 대상/목표가 보안이 아닌 다른 프로토콜까지 확장되었다. 그러나, 이 프로토콜들에서는, 보안이 본질적 특성이다. 예를 들어, 전자 경매 프

로토콜은 대상 (지불을 위한 교환 상품)이 보안과 관련되어 있지 않다는 점에서 보안 프로토콜이 아니다. 그러나, 지불을 위한 상품 교환과는 별개로 상품/지불의 복제나 상품/지불의 미수령에 대한 잘못된 주장이 없을 것을 보장해야 하기 때문에, 보안은 중요하고 필요한 특성이다. 보안 프로토콜과는 달리, 이러한 기타 프로토콜들은 오로지 침입자를 모델링함으로써만 안전한 것으로 보여질 수 있다. 기술적으로, 프로토콜의 목표와 특성 사이의 구분이 필요하지 않으며 보안 특성이 공동으로 프로토콜의 목표/대상에 추가될 수 있다는 주장이 제기될지 모른다. 그러나, 그러한 접근법은 프로토콜 목표의 성가시고 오래된 정의를 낳으며, 과정 증명에서 많은 오류로 이어질 수 있다.

한 가지 대안적 접근법은, 침입자를 모델링하는 정리 증명자(theorem prover)를 이용하는 것이다 [5]. 그러나, 많은 정리 증명자들은 형식증명된 모형에 기초를 두고 있지 않다. 그러므로, 그 가정들은 정확성을 위해, 특히 분석되고 있는 애플리케이션에 대해 체크될 필요가 있다.

다양한 프로토콜(전자상거래 프로토콜과 같은) 보안의 형식 증명에 대한 폭넓은 연구의 부족으로 인해, 검증 논리는 어떤 종류의 침입자 모형을 구체화하지 못했다. 아래에서 모든 가능한 침입자 행동을 열거함으로써 침입자를 설명한다. 여기서 설명된 침입자는 정리 증명자 [5, 10]에서 모델링된 모든 행동을 할 수 있으며, 따라서, 논문에서 모델링된 다른 침입자들보다 더 강력하다. 우선, 수동적 침입자의 행동을, 그리고 나서는 능동적 침입자의 행동을 설명한다.

침입자의 가능한 수동적 행동은 다음을 포함한다.

· 메시지 배달을 방해하지만, 불명확하진 않다. 따라서, 반복적으로 보내진 메시지는 결국 수신자에게 도달할 것이다.

· 메시지를 가로챈다.

· 판독가능한 (plain text 또는 해독된) 메시지를 구성요소로 해체한다.

· 암호화 키가 침입자에게 알려진 메시지 (및 메시지 구성요소)를 해독한다.

· 반복된 해독 및 해체로부터 유도된 메시지 및 메시지 구성요소를 저장한다.

네트워크 상에서 메시지의 손실 또는 변조 (corruption)는 단순히 침입자에 의한 메시지 방해로 모델링될 수 있음에 유의해야 한다.(이 경우, 침입자를 네트워크 그 자체가 될 것이다).

침입자의 가능한 능동적 행동은 다음을 포함한다:

· 다음으로부터 메시지를 구축한다

저장된 메시지 및 메시지 구성요소
침입자에게 알려진 키
평문

· 구축된 메시지를 네트워크에 삽입한다.

재생 공격은 저장된 메시지로부터 구축된 메시지이며 따라서 위의 행동에 포함된다는 점에 유의해야 한다.

트랜잭션의 일차 실행의 초기에, 침입자는 어떤 "내부" 정보도 갖고 있지 않다. 침입자는 (자신의 개인 키의 가능한 예외와 함께) 공개적으로 이용가능하지 않은 정보를 지니고 있지 않다. 형식적으로, I가 침입자를 지칭한다면 B는 프로토콜에서 상인이다.

$B_I = \{\varnothing : \varnothing\} \cup \{ b\ public\_key\_of\ B,\ I\ public\_key\_of\ I,\ \frac{1}{i}\ private\_key\_of\ I\};$

$O_I = \varnothing;$

$S_I = \varnothing;$

$K_I = \{ \frac{1}{i}, b, i\};$

3.2.3 시스템의 초기 상태

프로토콜의 첫 번째 예의 시작에서의 시스템의 상태가 아래에서 주어진다. 트랜잭션에서 하나 이상의 고객, 상인 또는 침입자가 있다면, 모든 고객, 상인, 침입자의 초기 상태는 아래에서 보여지듯이 각각 고객 (A), 상인 (B), 침입자 (I)의 그것과 유사하다.

$B_A = \{\varnothing : \vdash \varnothing\} \cup$
$\{b\ public\_key\_of\ B,\ i\ public\_key\_of\ I,\ a\ public\_key\_of\ A,\ \frac{1}{a}\ private\_key\_of\ a\}\ ;$

$B_B = \{\varnothing : \vdash \varnothing\} \cup \{b\ public\_key\_of\ B,\ i\ public\_key\_of\ I,\ \frac{1}{b}\ private\_key\_of\ B\};$

$B_I = \{\varnothing : \vdash \varnothing\} \cup \{b\ public\_key\_of\ B,\ i\ public\_key\_of\ I,\ \frac{1}{i}\ private\_key\_of\ I\};$

$O_A = \varnothing;\ O_B = \varnothing\ ;\ O_I = \varnothing;$

$S_A = \varnothing;\ S_B = \varnothing\ ;\ S_I = \varnothing;$

$K_A = \{ \frac{1}{a}, a, b, i\};$

$K_B = \{ \frac{1}{b}, b, i\};$

$K_I = \{ \frac{1}{i}, b, i\};$

모든 참여자(침입자를 포함한)는 동의어반복(tautologies), 키를 공개적으로(여기서는, 상인 그리고 선택적으로는 고객/침입자) 밝히기를 선택한 이들의 개인 키 및 공개 키만을 알고 있다. 또한, 유지되는 모든 믿음은 참이다. 그러므로, R(A)는 rightly_believes에 의해 대체되는 believes 연산자를 지닌 A와 동일하다. 참가자 중 누구도 어떤 것을 말하거나 듣지 않았다.

3.2.4 수동 공격 하에서의 보안 분석

수동 공격 동안에, 침입자는 어떤 메시지도 네트워크에 삽입할 수 없다. 그러나, 침입자는 메시지를 방해하고 배달을 일시적으로 막을지 모른다. 따라서, 프로토콜의 어떤 단계 이후(프로토콜이 방해시 종결된다면) 또는 단계의 어떤 수만큼의 반복 이후 (침입자가 더 이상 방해하지 않을 때까지 프로토콜 단계가 재시도된다면), 시스템이 안전하다는 것을 보여주는 일이 필요하다.

프로토콜 a가 단계 $a_1, a_2, ..., a_n$으로 이루어진다고 하자. 프로토콜이 방해로 종결된다면, 다음을 보여주어야 한다: $\forall\ ^{n}_{i=1}\{R[A]a_1, a_2, ..., ai\{R[C]\}$.

다시 말하면, 프로토콜은 각 단계 이후 안전하다. 송신자에 의해 승인이 수신될 때까지 프로토콜 단계가 반복된다면, 다음을 보여주어야 한다: $\forall\ ^{n}_{i=1}$ {R[C] $ai\ \{R[C]\}$.

다시 말하면, 단계의 어떤 수만큼의 반복이 여전히 시스템을 안전한 상태로 놓아둔다. 위의 두 진술로부터, 프로토콜에서 n 단계가 있다면 프로토콜이 n번만큼 분석되어야 한다는 것이 명백하다. 그러나, 프로토콜의 어떤 단계 i에 대해 어떻게 프로토콜을 분석하는가? 위의 두 진술이 모두 초기 상태 (A 또는 C), 필요한 최종 상태 (C), 그리고 초기 상태와 최종 상태 사이에서 일어나는 하나나 그 이상의 프로토콜 단계를 명시한다는 점에 유의해야 한다. 정리 1로부터, 만약 다음과 같다면 {R(A)} a {R[C]}임을 알고 있다.

$\wedge\ A \cup T(a)$ *is positive*
$\wedge\ A$ *allows* a
$\wedge\ A \cup T(a) \vdash C$

위의 세 결합이 참임을 보여주기 위해 정의1, 정의3, 정의4의 *T(a), positive, allows*의 정의를 이용한다.

3.2.5 능동 공격 하에서의 보안 분석

능동 공격 동안에, 침입자는 위조된 메시지를 네트워크로 삽입한다. 그러나, 이러한 메시지는 최소한 참여자 중 하나가 그 메시지를 적절한 프로토콜 단계라고 잘못 판단할 수 있을 때에만 의미있는 공격을 구성한다. 따라서 능동적 침입자에 의한 행동의 무한 공간을 잠재적으로 유해할 수 있는 행동의 유한한 집합으로 줄인다.

형식적으로, 단계 a_j가 $P \rightarrow Q : X^k$를 구성한다면, 침입자 행동 i_j는 형태 $I \rightarrow Q : x^k$가 되어야 한다. 여기서, x는 X와는 구별되지만 그것이 X대신 프로토콜 메시지 내용으로 기록되기 위해서는 동일한 형태이다.

프로토콜이 능동 공격 하에서 안전하다는 것을 보여주기 위해서, 침입자가 프로토콜의 어떤 단계에서도 오해의/유해한 메시지를 보낼 수 없다는 것을 보여주어야 한다. 형식적으로, 다음을 보여주어야 한다:

$$\forall\ ^{n}_{j=1} A \cup T(a_1, a_2, \cdots, a_j)\ \textit{not allows}\ i_j$$

위의 침입자 행동이 유효하지(또는 가능하지) 않다는 것을 보여주기 위해서 *allows*의 정의를 이용한다.

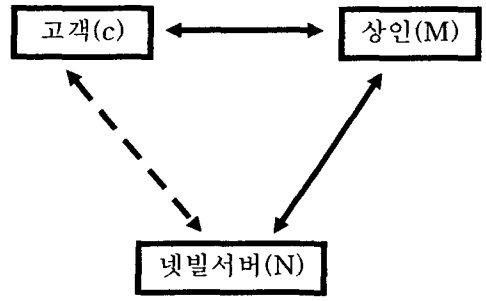
4 보안의 형식 증명으로 구조 설명하기

구조를 설명하기 위해서 잘 알려진 전자상거래 프로토콜 NetBill[9]을 선택한다. 프로토콜의 관련 당사자들은 고객 C, 상인 M, NetBill 서버 N이다. 넷빌 거래에서, 고객과 상인은 첫 두 단계에서 서로 상호작용하며, 넷빌 서버는 판매자가 거래 요청을 제출하는 지불 단계가 지난 이후 관련된다. 고객은 통신 실패의 경우 또는 관리자 기능을 요청할 때에만 넷빌 서버와 직접적으로 접촉한다.

<그림 1>은 넷빌 거래에서 당사자간의 관계를 보여주며, 이 장에서 구조가 주어진 프로토콜의 보안을 증명하기 위해 어떻게 적용될 수 있는지를 설명하기 위해 이 프로토콜의 단계 1과 단계 5를 위한 보안 증명을 제시한다. 아래에서 프로토콜의 간단한 개요를 제시한다. X가 명시된 메시지 Msg를 Y에게 보낸다는 것을 나타내기 위해 "X → Y Msg"라는 표기법을 이용한다.

NetBill 프로토콜은 다음의 단계로 구성된다.

가. C⇒M 가격 요청
나. M⇒C 견적
다. C⇒M 상품 요청
라. M⇒C 키 K로 암호화된 상품
마. C⇒M 서명된 EPO
바. M⇒N 배서된 EPO (K 포함)



Transaction Protocol

Auxiliary Message

<그림 1> 넷빌 모델

사. N⇒M 서명된 결과 (K 포함)
아. M⇒C 서명된 결과 (K 포함)

프로토콜 단계 1과 2의 교환에 의해 고객과 판매자 사이에는 제안 및 수락 협상 단계의 목표가 달성되며, 단계 5에서 생성된 EPO가 단계 6에서 수락되기 전 고객이 넷빌에 인증되어야 하므로, 인가된 고객만이 넷빌 계정에 대해 요금을 청구할 수 있다.
넷빌 프로토콜은 이러한 목표를 달성하기 위해서 강력한 인증 및 프라이버시, 원자적 지불 및 배송 프로토콜, 탄력적인 접근 통제 시스템을 제공한다.

4.1 수동 공격 하에서의 보안 분석

단계 1이후 보안 증명이 아래에서 제시된다. 이것은 수동 공격만이 고려되는 증명의 첫 번째 단계이다. 다음에서 능동 공격이 어떻게 증명으로 구체화되는지를 보여준다. 본 논문에서의 증명은 가독성을 높이기 위해 증명의 개요를 기술하고 개요에서 한 단계에서 다음 단계로의 전이가 명확하지 않은 곳에서는, 유도된 단계의 증명이 개별적으로 주어진다.

단계 1을 위한 증명
증명 : {A} a_1 {C}
증명 개요: 시스템이 안전하다는 것 또는 *{A} a_1 {C}*를 증명하기 위해 정리 1을 이용한다.

a_1 : $C \rightarrow M$ *Price request*
- $A \cup T(a_1)$ *is positive*
- *A allows* a_1
- $A \cup T(a_1) \vdash C$

증명 단계 1.1
증명 : $A \cup T(a_1)$ is positive
정의 4를 사용하여 증명
$A \cup T(a_1)$ *is positive*
- $T(a_1)$:= $\wedge$ *C once_said Price request*
  $\wedge$ *M sees Price request*
  $\wedge$ *I sees Price request*
- $T(a_1)$ *is positive*
- *A is positive*

증명 단계 1.2
증명 : A allows $T(a_1)$
정의 3을 이용하여 증명
A allows $T(a_1)$
- *A* $\vdash$ *C believes Price request*

증명 단계 1.3
증명 : $A \cup T(a_1) \vdash C$
증명 개요: 단계 a_1에서 어떤 원치 않는 정보도 침입자에게 노출되지 않았음을 보여준다.

$A \cup T(a_1) \vdash C$
- $T(a_1)$:= $\wedge$ *C once_said Price request*
  $\wedge$ *M sees Price request*
  $\wedge$ *I sees Price request*
- *New state is A* $\cup \wedge$ *Price request* $\in O_C$
  $\wedge$ *Price request* $\in S_M$
  $\wedge$ *Price request* $\in S_I$
- $\wedge \frac{1}{a} \notin S_I$
  $\wedge \frac{1}{b} \notin S_I$
  $\wedge$ *payment* $\notin S_I$
  $\wedge$ *product* $\notin S_I$

4.2 능동 공격 하에서의 보안 분석

능동적인 침입자 공격을 분석할 때에는 침입자가 프로토콜 메시지처럼 판독되지만 프로토콜 메시지와 동일하지 않은 메시지를 보내도록 허용되진 않는다는 점을 보여주어야 한다. 단계 1이 가격 요청이므로 가격 요청처럼 판독되는 어떤 메시지는 단계 1의 정확한 복제품이어야 한다. 따라서, NetBill 프로토콜의 단계 1을 위한 증명의 능동적 침입자 부분은 무의미하게(vacuously) 참이다.

실제 참여자를 속이기 위해 내용이 한계적으로 바뀔 수 있는 그러한 메시지에 대해서, 침입자가 변경된 메시지를 보내도록 허용되지 않았음을 보여주

기 위해 정의 3을 이용하며, 단계 5를 위한 능동 공격 하에서의 보안을 증명함으로써 이를 설명한다. 단계5는 프로토콜의 중간에 해당하고, 능동 공격의 관점에서 중요한 단계이므로(고객이 실제로 주문을 하는 단계이며, 어떤 침입자도 고객인 것처럼 주문을 할 수 있도록 되기를 원하지 않는다) 선택한다.

단계 5를 위한 증명

가정 : $A \cup T(a_1, a_2, \cdots, a_4) \vdash C$

수동적 공격하 $A \cup T(a_1, a_2, \cdots, a_5) \vdash C$

증명 : $A \cup T(a_1, a_2, \cdots, a_5)$ not allows I_5

증명 개요: 단계 4에 대해 보안이 이미 증명되었으며 단계 5에 대해서는 수동 공격 하의 보안이 증명되었다고 가정한다. 능동 공격 하의 보안을 증명하기 위해 정의 3을 이용한다.

$a_5 : I \rightarrow M$ (modified EPO) $^{C's\ private\ key}$

· 정의 3으로부터 , A allows $a \rightarrow Q : X^k$, $k \notin K_p$

:= $A \vdash P$ sees X^k

· $A \cup T(a_1, a_2, \cdots, a_5)$ allows $I \rightarrow M$: (modified EPO) $^{C's\ private\ key}$,

C's private key $\notin K_I$

:= $A \cup T(a_1, a_2, \cdots, a_5) \vdash I$ sees (modified EPO) $^{C's\ private\ key}$

· 가정으로부터 C's private key $\notin K_I$

· $A \cup T(a_1, a_2, \cdots, a_5)$ allows $I \rightarrow M$: (modified EPO) $^{C's\ private\ key}$

:= $A \cup T(a_1, a_2, \cdots, a_5) \vdash I$ sees (modified EPO) $^{C's\ private\ key}$

· I not sees (modified EPO) $^{C's\ private\ key}$

증명 : I not sees (modified EPO) $^{C's\ private\ key}$

증명 개요; 이단계의 증명은 BAN 로직의 공리를 이용한다.

I not sees (modified EPO) $^{C's\ private\ key}$

· Good key ensures utterer 공리로부터

$\vdash$ k private_key_of P $\wedge$ R sees $[X]^k \rightarrow$ P once_said X

· $\vdash$ k private_key_of C $\wedge$ I sees (modified EPO)^k

$\rightarrow$ C once_said (modified EPO)

· C not once_said (modified EPO)

5 결론

지금까지 시스템의 보안을 확인하기 위한 다양한 방법들을 검토했다. 본 논문에서 사용한 지불프로토콜 분석도구인 BAN 로직은 프로토콜 모델을 위한 많은 가정을 필요로 하는 문제점은 있지만 프로토콜의 안전성 분석을 위한 토대를 마련했으며, 많은 개선된 로직이 개발되어 왔다.

본 연구에서는 BAN 논리를 위한 의미론을 제시했고 보안 프로토콜의 증명을 위해 BAN 논리의 확장을 개발했으며, 상태 트랜잭션을 위해 BAN 논리를 이용하여 대표적인 지불프로토콜인 NetBill 프로토콜에 대한 형식기법을 제안했다. 이러한 확장된 논리는 원자성(atomicity), 익명성(anpnymity) 및 프라이버시(privacy)와 같은 다른 중요한 특성의 증명을 위해 이용될 수 있다.

향후, 연구과제는 일반적인 쌍방(two party) 트랜잭션을 위한 트랜잭션 프로토콜을 개발하는 것이다.

<참고문헌>

[1] A.Bleeker and L. Meertens. "A Semantics for BAN Logic", In Proceedings of the DIMACS Workshop on Design and Formal Verification of Security Protocols, 1997.

[2] Benjamin Cox, J. D. Tygar, Marvin Sirbu, "NetBill Security and Transaction Protocol", http://www.ini.cmu.edu/NETBILL/pubs/Usenix.html #HDR2

[3] C. Meadows. "Analyzing the needham-Schroeder Public Key Protocol: A Comparison of Two Approaches", In Proceedings of ESORICS, 1996.

[4] E. Snekkenes. "Exploring the BAN Approach to Protocol Analysis", In Proceedings of the IEEE Symposium on Research in Security and Privacy, 1991.

[5] F.Germeau and G. Leduc. "Model-based Design and Verification of Security Protocols using LOTOS", In Proceedings of the DIMACS Workshop on Design and Formal Verification of Security Protocols", 1997.

[6] http://www.cerberussystems.com/INFOSEC/stds/fip140-1.htm

[7] http://csrc.nist.gov/publications/nistpubs/800-9/800-9.pdf

[8] http://csrc.nist.gov/publications/nistpubs/800-2/800-2.txt

[9] J.D. Tygar. "Atomicity in Electronic Commerce", PODC, 1996.

[10] L.C. Paulson. Isabelle, "A Generic Theorem Prover", Lecture Notes in Computer Science, 1994.

[11] M.Abadi, and R. Needham. "Prudent Engineering Practice for Cryptographic Protocols", In Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, May 1994.

[12] M.Burrows, M.Abadi, and R. Needham. "A Logic of Authentication", ACM Transaction on Computer Systems, 1990.

[13] M.Abadi, Mark R. Tuttle, "A Semantics for a Logic of Authentication", In Proceedings of the Tenth Annual ACM Symposium on Principles of Distributed Computing, Auguest, 1991.

[14] NetBill: An Internet Commerce System Optimized for Network Delivered Services ,http://www.ini.cmu.edu/netbill/pubs/CompCon.html #RTFToC4

[15] P. Syverson. "Adding Time to a Logic of Authentication", In Proceedings of the First ACM Conference on Computer and Communications Security, 1993.

[16] S. Subramanian. "Design and Verification of Protocols for Secure Transaction Execution in Electronic Commerce", Dissertation Proposal, OSU-CISRC-10/98-TR40, 1998.

[17] S. Subramanian and M Singhal. "Design and Verification of a Secure, Atomic Transaction Execution Protocol for Electronic Commerce", Technical Report, OSU-CISRC-10/98-TR12, 1998.

[18] S. Subramanian and M Singhal. "Protocols for Secure, Atomic Transaction Execution Protocol in Electronic Commerce", Technical Report, OSU-CISRC-10/97-TR49, 1997.

[19] S. Subramanian and M Singhal. "A Methodology for Detecting Violation of Real-Time Constraints in Secure Electronic Commerce Transactions", Technical Report, OSU-CISRC-5/98-TR15, 1998.