情報保護學會論文誌
第13券 第5號, 2003. 10

인증 및 키 분배 프로토콜의 논리성 검증을 위한 ASVO 로직*

권 태 경*, 임 선 간**, 박 해 룡**

Automation-considered SVO Logic for Verifying Authentication and Key Distribution Protocols

Teakyoung Kwon*, Seongan Lim**, Haeryong Park**

요 약

본 논문에서는 인증 및 키 분배 프로토콜에 대한 논리성 검증을 위한 ASVO 로직을 제안한다. ASVO 로직은 기존의 인증 로직 중 하나인 SVO 로직에 대해서 자동 검증을 고려하여 변형 설계한 로직이다. ASVO 로직은 구문적/의미적 구조의 안전성을 갖는 로직으로서, 비교적 간소화된 증명 단계를 갖는다. 또한 Isabelle/Isar 시스템을 이용하여 구현된 Isabelle/ASVO 시스템은 ASVO 로직을 통한 반자동 검증을 지원한다.

ABSTRACT

This paper presents the ASVO (Automation-considered SVO) Logic that can be used for verifying authentication and key distribution protocols. The ASVO logic was designed for automatic verification, in a way to modify the SVO logic, one of the most famous authentication logics. The ASVO logic is syntactically and semantically sound, and requires relatively simple verification steps. Also we implemented the Isabelle/ASVO system which supports semi-automated verification, by using the Isabelle/Isar system.

keyword :

Ⅰ. 서 론

암호프로토콜의 논리적 안전성 검증 방법은 공격 구성(attack construction) 방법과 추론 구성(inference construction) 방법으로 크게 나누어 볼 수 있다.[6] 우선 공격 구성 방법은 프로토콜을 상태 기계(state machine)들의 상호 작용으로 간주하고 모든 가능한 상태들에 대한 탐색을 통해서 불안전한 상태에 도달하는 구체적인 공격 경로를 찾도록 하는 방법을 일컫는다. 추론 구성 방법은 modal 로직에 바탕을 두며, 지식(knowledge) 또는 신뢰(belief) 분석을 통해서, 논리적 안전성에 관련된 목적을 정확하게 이루는지 구체적으로 추론하는 방법을 일컫는다. 따라서 추론이 불가능한 경우, 추론 진행을 위해서 필요한 부가적인 요구사항을 통하여 프로토콜의 약점을 찾을 수 있다.

본 논문에서는 추론 구성 분야에서 대표적인 기법 중 하나인 SVO 로직에[11,12] 바탕을 둔 ASVO (Automation-considered SVO) 로직을 제안한다.

* 본 연구는 한국정보보호진흥원에서 지원하는 위탁과제로 수행하였습니다.(과제번호 2002-S-073)
** 세종대학교 컴퓨터공학부(tkwon@sejong.ac.kr)
*** 한국정보보호진흥원({seongan, hrpark}@kisa.or.kr)

1.1 SVO 로직

1994년 Syverson과 van Oorschot가 제안한 SVO 로직은, modal 로직 이론에 바탕을 둔 기존의 BAN 로직, GNY 로직, AT 로직, 그리고 vO 로직을 통합하여 보다 효과적이고 강력한 로직의 완성을 추구한 것이다.[1,2,5,11,12,14] 즉, 기존 신뢰 기반 로직(doxastic logic)들의 장점을 취합하였으며, 20개 정도의 간략한 공리 스키마로 재구성하였다. SVO 로직은 비록 구문적/의미적으로 안정적인 것으로 잘 알려져 있지만, 프로토콜 증명을 위한 이상화(idealization) 과정이나 메시지의 해석 부분에서 여전히 검증자의 전문 지식과 프로토콜에 대한 정확한 해석을 필요로 하였다. 예를 들면, A received $\{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$와 같은 구문에 대하여, 다음과 같이 이상화하는 과정을 필요로 했다.

$$A \text{ believes } (A \text{ received } \{N_A, B, *_1, *_2\}_{K_{AS}}$$
$$\Rightarrow A \text{ received } \{N_A, B, A \xleftrightarrow{K_{AB}} B, fresh(K_{AB}), *_2\}_{K_{AS}})$$

즉, 이미 이상화 단계에서 검증자는 주체 A의 메시지 이해($*$)와 해석($A \xleftrightarrow{K_{AB}} B$, $fresh(K_{AB})$)에 대하여 직접 분석해야 했으며, 향후 검증과정에서도 $*_2$와 같이 미확인된 부분을 직접 복원하거나, 어떤 메시지에 대해서는 특별히 공리 적용[1]을 위한 재해석을 추가적으로 직접 하여야 했다. 이로 인하여 추론 과정이 검증자의 전문성에 의존적일뿐만 아니라, 결과적으로 정확한 추론을 위한 자동 검증 도구를 만들기도 까다롭다. 현재까지 SVO 로직 자동 검증과 관련된 연구로는 SVD 로직을 구현한 C3PO가 유일하다.[3] 하지만 Isabelle/Pure를 사용해서 구현된 SVD 로직의 경우, SVO 로직을 70여개 이상의 규칙으로 단편화하였다는 면에서 근본적으로 상당한 거리가 있다. 또한 공리적용을 위한 메시지 해석은 여전히 검증자에 의해서 이루어져야만 했다.

1.2 ASVO 로직

ASVO 로직은 SVO 로직에 바탕을 두지만, 프로토콜 증명 단계를 간소화하며, 자동 검증을 지원하여 정확한 추론 경로를 쉽게 찾을 수 있도록 하는 것을 주요 목적으로 한다. 따라서 SVO 로직과 같이 단순한 로직 구조를 유지하지만, 프로토콜 검증 과정에서 검증자의 수동적인 해석을 요구하지 않으며, SVD 로직과 같이 자동 검증을 지원하지만, 오히려 SVO 로직의 구조적 단순성(simplicity)을 유지하도록 한다. 이와 같은 ASVO 로직의 구조적 특징을 간략히 요약하면 다음과 같다.

1) SVO 로직의 단순한 구문 구조와 함께, 의미적 기본 구조를 최대한 유지하였다.
2) SVO 로직에서 CA와 ISA 규정 단계를 제거하였으며, 그대신 이와 같은 부분을 논리적으로 해석할 수 있는 공리 스키마들을 정의하였다. 여기에는 received2, from, vague, understands와 같은 새로운 predicate들이 함께 추가되었다.
3) 구현을 위해서 필요한 연산자와 변환 함수를 정의하였다. 예를 들면, 각 formula의 연결과 term에 대한 쌍 구성을 위한 연산자와 formula-message 변환 함수를 정의하였다.
4) 메타 표기로 인한 모호성 제거를 위해서, 함수의 명확한 정의나 규칙을 포함하도록 하였다. 예를 들면, 역함수를 갖는 일반 함수 F()와, 일방향 해쉬 함수 H(), Diffie-Hellman 함수 F0(), 그리고 MAC H(K)에 대한 정의를 하였으며, 이에 관련된 공리 스키마들 역시 각각 정의하였다.
5) 무한 순환 발생을 피하기 위하여 predicate에 대한 구문 및 의미적인 선형 구조를 구성하였다.
6) Gödel의 두 번째 공리에 바탕을 두고, 후향 증명을 하도록 하였다. 따라서 P believes $\varphi \Rightarrow P$ believes ψ와 같은 추론 증명을 위해서 P believes ($\varphi \Rightarrow \psi$)에 대한 추론을 할 수 있다.

한편 제안된 ASVO 로직을 반자동 추론을 지원하는 Isabelle/Isar 시스템을 통하여 구현하였으며,[10] 구현된 Isabelle/ASVO 시스템을 통하여 다양한 유형의 프로토콜에 대한 안전성 검증[2]과 갱신을 하였다. 특

1) 예를 들면, 참고문헌 [12]에서 SVO 공리 Ax3 적용을 위해서 X_K로부터 $X^Q{}_K$를 해석 유도함.

2) 형식 논리에 대한 완전 자동화는 주어진 목표(goal)에 대한 추론을 자동화하는 것을 의미한다. 각 프로토콜은 여러 가지 목표를 가지며, 그 목표 또한 추론을 위해서 부목표(subgoal)로 나누어질뿐만 아니라, 결과적으로 이에 대한 사용자의 개입이 요구되므로, 오히려 사용자의 관점에서는 반자동화(semi-automation)라고 볼 수 있다.

히 Isabelle/ASVO 시스템에 대해서는 선행 연구[16]에서 이미 소개한 바 있으므로, 본 논문에서는 ASVO 로직에 대해서 자세히 다루도록 한다. 먼저 II장에서는 ASVO 로직의 논리 언어에 대해서 기술하고, III장에서는 ASVO 로직의 구문구조를 설계한다. IV장에서는 ASVO 로직의 의미구조에 대한 검증을 하고, V장에서는 ASVO 로직 검증 방법에 대해서 살펴본다. 마지막으로 VI장에서는 Isabelle/ASVO 도구를 이용한 검증 결과에 대해서 기술하고, VII장에서 결론을 맺는다.

II. 논리 언어(logic language)

ASVO로직의 언어는 이미 BAN 로직, AT 로직, SVO 로직 등의 선조격 인증 로직에서 정의된 de facto적 언어 구조와 표기법을 갖는다.[1,2,11] ASVO 로직의 논리 언어는 프로토콜의 메시지(message)를 위한 언어와 논리식(formula)을 위한 언어로 구분한다. 특히 이것은 SVO 로직에 기반[3]을 두고 이루어졌으며, 따라서 언어에 대한 서술은 많은 유사한 점이 있다. 주체, 공유키, 공개키, 개인키 등의 상수 심볼로 이루어진 항(term) 집합 T_0와 함께 원시 명제 상수를 포함하는 집합 T를 정의하고, T에서 메시지와 논리식을 위한 언어를 정의한다.

2.1 메시지(message)

메시지에 대한 형식 언어 M_T는 원시 항들의 집합 T에서 비롯된 기본 언어이며 다음을 만족한다.

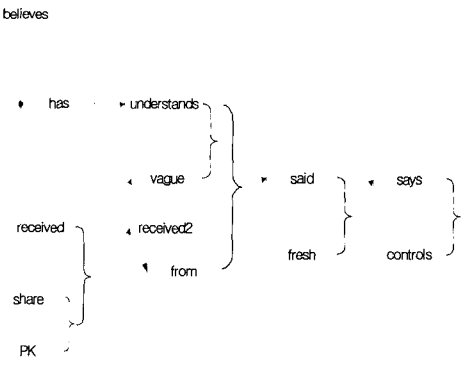
1) 만약 $X \in T$ 이면, X는 메시지이다.
2) $X_1, \ldots, X_n$가 각각 메시지이고 F가 어떤 함수이면, $F(X_1, \ldots, X_n)$은 메시지이다. 즉, 메시지의 연결자는 콤마(,)이다.
3) φ가 논리식이면, φ는 메시지이다.

3) ASVO 로직은 predicate들에 대한 정의를(예를 들면, received2, from, understands,vague 등) 새로이 하고, 메시지 집합에 대한 모델링을 새로이 하여, 결과적으로 SVO 로직과 명확히 구분된다. III절에서 정의되는 로직의 논리 구문 구조를 살펴보면, 기본적인 Believing 스키마를 제외한 나머지 논리 스키마들이 다르게 정의됨을 알 수 있으며, IV절의 의미 구조도 달라짐을 알 수 있다.

2.2 논리식(formula)

논리식 F_T 역시 다음을 만족하는 기본 언어이다. [그림 1]은 ASVO 로직의 각종 predicate들에 대한 관계도를 나타낸다.

1) p가 원시 논리식이면, p는 논리식이다.
2) φ와 φ'가 각각 논리식이면, $\neg\varphi$와 $\varphi \wedge \varphi'$도 모두 논리식이다. 즉, 논리식의 연결자는 논리곱을 의미하는 $\wedge$이다.
3) $\Rightarrow$는 논리식의 implication을 의미한다. 즉, $\varphi \Rightarrow \psi$이면 φ의 진리값이 참일 경우 ψ의 진리값도 참이다.
4) P가 주체이고 φ가 논리식이면 다음은 모두 논리식이다. 여기서 believes는 modal 연산자에 해당한다.
   i) P believes φ
   ii) P controls φ
5) P가 주체이고 X가 메시지이면 다음은 모두 논리식이다.
   i) P has X
   ii) P received X
   iii) P received2 X
   iv) X from P
   v) P said X
   vi) P says X
   vii) P understands X
6) X가 메시지이면 다음은 모두 논리식이다.
   viii) fresh(X)
   ix) vague(X)



[그림 1] ASVO 로직 predicate 관계도

7) P와 Q가 주체이고 K가 키이면 다음은 모두 논리식이다.

i) $P \xleftrightarrow{K} Q$

ii) $PK(P,K)$

2.3 키 표현

ASVO 로직에서 $PK(P,K)$는 주체 P의 공개키 K를 의미하며 다음과 같이 각각 그 소유와 의미가 정리된다.

1) $PK_{\phi}(P,K)$: 암호화를 위한 공개키
2) $PK_{\sigma}(P,K)$: 서명 확인을 위한 공개키
3) $PK_{\delta}(P,K)$: 키 분배를 위한 공개키

한편, K^{-}는 K의 역수를 의미한다. 따라서 공개키 K에 대해서 개인키는 K^{-1}이며, 대칭키 K에 대해서는 $K = K^{-}$이다. 이것은 키에 대한 공리 정의를 단순하게 만든다. 이 개념을 함수에 대해서 일반화시키면 F와 F^{-}를 얻을 수 있다. 또한 K_{PQ}와 같이 주체를 대문자로 표기하는 경우는 대칭키를, 그리고 K_p, K_q, K_{pq}와 같이 주체를 소문자로 표기하는 경우는 Diffie-Hellman 키를 의미한다. 즉, Diffie-Hellman 키 변환 함수를 $F_0()$라고 정의[4]할 때 $K_{pq} = F_0(K_p, K_q)$와 같이 표현할 수 있다.

2.4 정리 증명

$\vdash$는 메타논리적 심볼로서 $\Gamma \vdash \varphi$는 ASVO로직을 통하여 논리식 집합 Γ로부터 논리식 φ가 유도될 수 있다는 것을 의미하며, $\vdash \varphi$는 theorem을 의미한다. 따라서 일상적으로 $\Gamma \vdash \varphi$를 $\vdash \Gamma \Rightarrow \varphi$와 같은 theorem으로 나타내고 증명할 수 있다.

Ⅲ. 논리 구문 구조(logic syntax)

3.1 추론 규칙

ASVO로직은 AT로직이나 SVO로직과 마찬가지로 가장 기본적인 추론 규칙, 즉 Modus Ponens와 Necessitation을 명확히 하고 이를 기반으로 각 공리 스키마를 정의한다.[1,11,15]

MP) φ 와 $\varphi \Rightarrow \psi$에서 ψ를 추론한다.

Nec) $\vdash \varphi$ 에서 $\vdash P$ believes φ를 추론한다.

스키마 구성도 AT로직이나 SVO로직으로부터 유도되었지만 근본적으로 많은 차이가 있다.

3.2 Believing 공리 스키마

ASVO 로직은 believes를 modal 연산자로 갖는 doxastic 로직이다. 따라서 다음과 같이 신뢰에 관한 기본적인 공리 스키마를 갖는다.

A1) P believes $\varphi \wedge P$ believes $(\varphi \rightarrow \psi)$
$\Rightarrow P$ believes ψ

A2) P believes $\varphi \Rightarrow \varphi$

A3) P believes $\varphi \Rightarrow P$ believes (P believes φ)

A4) $\neg$(P believes φ)
$\Rightarrow P$ believes ($\neg P$ believes φ)

이와 같은 공리 스키마는 modal 로직의 기본적 공리들이며, 관련 로직에서 공유한다.[21] 특히 ASVO 로직에서 Believing 공리 스키마를 제외한 모든 공리 스키마는 주체가 신뢰하는 사실에 대한 증명을 위한 것이다. 즉, $\vdash P$ believes $\varphi \Rightarrow P$ believes ψ의 증명을 위해서 $\vdash P$ believes $(\varphi \Rightarrow \psi)$에 대한 증명을 할 수 있다. 이것은 Gödel의 두 번째 공리에 기반하며, SVO로직과 근본적으로 구별된다.[15]

3.3 Source Association 공리 스키마

ASVO 로직은 수신한 메시지가 식별 가능할 경우에만, 송신자와 그 송신 내용에 대한 확신을 할 수 있도록 한다. 따라서 대칭키를 사용할 경우나, 전자서명을 사용하는 경우에 한하여 다음과 같은 공리 스키마를 갖는다.

A5) P understands $X \wedge P$ received2 X from Q
$\Rightarrow Q$ said $(X, P \xleftrightarrow{K} Q)$

A5a1) ($PK_{\sigma}(Q,K) \wedge P$ received $\{X\}_{K^{-1}}$)
$\Rightarrow Q$ said $X \wedge P$ received X

4) 우리는 이와 같은 지수 계산 함수 $Fi()$의 보다 구체적인 정의를 통해서, 향후 Diffie-Hellman variant들의 분석도 가능할 것으로 기대한다.

송신을 위한 predicate으로 received와 received2를 구분한 근본적인 이유는, 메시지 소유와 이해에 대한 분석을 명확히 하기 위하여, 논리식 유도의 순환성(circularity)을 제거도록 한 것이다. [그림 1]을 참고하라.

3.4 Key Agreement 공리 스키마

ASVO 로직은 SVO 로직과 VO 로직을 바탕으로 키 합의에 대한 공리 스키마를 갖는다.

A6) $(\mathrm{PK}_{\delta}(\mathrm{P},\mathrm{K}_{\mathrm{p}}) \wedge \mathrm{PK}_{\delta}(\mathrm{Q},\mathrm{K}_{\mathrm{q}})) \Rightarrow P \xleftarrow{K_{pq}} Q$

A6a1) P says $K_p \wedge \neg \mathrm{PK}_{\delta}(R,K_p) \Rightarrow \mathrm{PK}_{\delta}(P,K_p)$

A7) $\varphi F_0(K_p,K_q) \equiv \varphi K_{pq}$

A7a1) $(\mathrm{PK}_{\delta}(\mathrm{P},\mathrm{K}_{\mathrm{p}}) \wedge P$ has $K_q) \Rightarrow P$ has $F_0(K_p,K_q)$

특히 A6a1)과 같이 Diffie-Hellman 공개키의 근원지가 확인될 경우, 공개키의 소유자에 대해서 유도할 수 있다. 이것은 ASVO 로직이 기본적으로 주체의 신뢰에 대해서만 증명하는 것이기 때문에 가능하다. 즉, 예를 들어서 Q believes P says K_p와 Q believes P says K_q의 경우에 각각 $\mathrm{PK}_{\delta}(\mathrm{P},\mathrm{K}_{\mathrm{p}})$와 $\mathrm{PK}_{\delta}(\mathrm{P},\mathrm{K}_{\mathrm{q}})$를 유도할 수 있지만, 이것은 Q가 신뢰하게 되는 사실을 의미할 뿐 유도된 사실이 실제로 반드시 참이라는 것을 의미하지는 않는다. 즉, Q는 $\mathrm{PK}_{\delta}(\mathrm{P},\mathrm{K}_{\mathrm{q}})$와 같이 실제와 다른 사실을 신뢰할 수도 있지만, 오히려 이와 같은 그릇된 신뢰를 통해서 프로토콜의 약점을 확인할 수 있다. 한편 A7)의 경우, 논리식 φ에 대한 메타 공리 역할을 하게 되는데, 예를 들면 다음과 같은 공리들을 유도할 수 있다. 다음은 실제로 Isabelle/ASVO 시스템에서 사용되는 A7)의 인스턴스들이다.

A7i1) R has $F_0(K_p,K_q) \equiv R$ has K_{pq}

A7i2) fresh($F_0(K_p,K_q)$) $\equiv$ fresh(K_{pq})

3.5 Receiving 공리 스키마

ASVO로직은 수신된 메시지의 인식성(recognizability), 즉 식별 가능 여부에 다한 확인을 논리적으로 다룬다. 따라서 다음과 같은 수신에 관한 공리 스키마를 갖는다.

A8) P received $(X_1,\ldots,X_n) \Rightarrow P$ received X_i

A9) (P received $\{X\}_K \wedge P$ has K^-) $\Rightarrow P$ received vague(X)

A9a1) P understands $X \wedge P$ received vague(X) $\Rightarrow P$ received X

A9a2) P received vague($\{X\}_K$) $\Rightarrow P$ received $\{\mathrm{vague}(X)\}_K$

A10) P received $H(X) \wedge P$ has $X \Rightarrow P$ received2 X

3.6 Possession 공리 스키마

ASVO 로직은 모든 수신된 메시지와 송신한 메시지에 대한 소유를 논리적으로 명확히 정의한다. 따라서 다음과 같은 소유에 관한 공리 스키마를 갖는다.

A11) P received $X \Rightarrow P$ has X

A12) P said $X \Rightarrow P$ has X

A12a1) P says $X \Rightarrow P$ has X

A13) P has $(X_1,\ldots,X_n) \Rightarrow P$ has X_i

A14) (P has $X_1 \wedge \ldots \wedge P$ has X_n) $\Rightarrow (P$ has $F(X_1,\ldots,X_n))$

3.7 Key ownership 공리 스키마

ASVO 로직은 키의 소유에 대한 공리 스키마를 갖는다. A15)와 같이 P가 논리식 $P \xleftarrow{K} Q$를 메시지로서 송신한다는 것은 구체적으로 P가 송신한 메시지로부터 암시적으로 유도된 메시지[5]라는 사실을 의미한다. 이것은 $P \xleftarrow{K} Q$가 반드시 참인 경우이며, 따라서 P has K 와 같은 논리식을 유도할 수 있다.

A15) P said $P \xleftarrow{K} Q \Rightarrow P$ has K

A16) $\mathrm{PK}_{\phi}(\mathrm{P},\mathrm{K}) \Rightarrow P$ has K^{-1}

A16a1) $\mathrm{PK}_{\sigma}(\mathrm{P},\mathrm{K}) \Rightarrow P$ has K^{-1}

3.8 Saying 공리 스키마

ASVO로직은 송신 메시지에 대한 공리 스키마를

5) 따라서, formula-message 유형 변환이 필요하다.

갖는다. SVO로직의 기본적인 구성과 마찬가지로 시점에 대해서 논리적으로 구성한다.

A17) P said $(X_1, \ldots, X_n) \Rightarrow P$ said X_i
A17a1) P says $(X_1, \ldots, X_n)$
$\Rightarrow P$ says $X_i \wedge P$ said $(X_1, \ldots, X_n)$

3.9 Freshness 공리 스키마

ASVO 로직은 메시지의 신규성에 관한 공리 스키마를 갖는다.

A18) fresh(X_i) $\Rightarrow$ fresh($X_1, \ldots, X_n$)
A19) fresh($X_1, \ldots, X_n$)
$\Rightarrow$ fresh(F($X_1, \ldots, X_n$))

3.10 Jurisdiction 공리 스키마

ASVO 로직은 관할에 관한 공리 스키마를 갖는다. 즉, 주체 P가 관할하는 사실에 대해서 직접 언급할 경우, 그 사실에 대한 신빙성을 규명하도록 한다.

A20) $(P$ controls $\varphi \wedge P$ says $\varphi) \Rightarrow \varphi$

3.11 Nonce-Verification 공리 스키마

ASVO 로직은 메시지의 송신에 대한 시점을 판단하기 위한 공리 스키마를 갖는다.

A21) (fresh(X) $\wedge P$ said $X) \Rightarrow P$ says X

3.12 Symmetric goodness 공리 스키마

ASVO 로직은 일반적인 대칭 공유키와 Diffie-Hellman 키 합의를 통한 공유키의 대칭성에 대한 공리 스키마를 갖는다.

A22) $P \xleftrightarrow{K} Q \equiv Q \xleftrightarrow{K} P$
A22a1) $P \xleftrightarrow{K_{pq}} Q \equiv Q \xleftrightarrow{K_{qp}} P$

3.13 Interpretation 공리 스키마

ASVO 로직은 메시지에 대한 논리적 해석을 위한 공리 스키마를 가지며, 이를 통하여 프로토콜의 복잡한 이상화 개념을 제거한다.

A23) $(P \xleftrightarrow{K} Q \wedge P$ received $\{X\}_K)$
$\Rightarrow P$ received2 X from Q
A23a1) $(P \xleftrightarrow{K} Q \wedge P$ received $H(K, X))$
$\Rightarrow P$ received2 X from Q
A24) $(R$ says $K \wedge R$ controls $P \xleftrightarrow{K} Q)$
$\Rightarrow R$ says $P \xleftrightarrow{K} Q$
A24a1) $(R$ says $K \wedge R$ controls fresh(K))
$\Rightarrow R$ says fresh(K)
A25) $(P \xleftrightarrow{K} Q \wedge$ fresh(K)) $\Rightarrow$ fresh($P \xleftrightarrow{K} Q$)

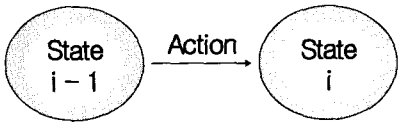
3.14 Recognition 공리 스키마

ASVO로직은 메시지의 인식성, 즉 식별 가능 여부에 대해서 논리적으로 정의한 공리 스키마를 갖는다. 이것은 소유에 대한 논리식으로부터 비롯되며, 수신 메시지에 대해서 명확한 판별을 할 수 있도록 한다.

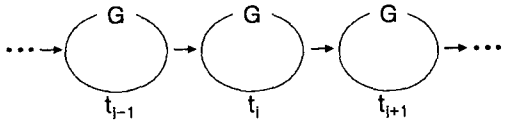
A26) P has $X \Rightarrow P$ understands X
A27) P understands X_i
$\Rightarrow P$ understands $(X_1, \ldots, X_n)$
A28) P understands $X \wedge P$ has K^{-}
$\Rightarrow P$ understands $\{X\}_K$
A28a1) P understands $X \wedge \mathrm{PK}_{\sigma}(\mathrm{Q}, \mathrm{K})$
$\Rightarrow P$ understands $\{X\}_{K^{-1}}$

SVD로직에서도 understands와 같은 이름의 predicate을 정의한다. 하지만 has와 동치 관계인 SVD로직의 understands는 ASVO로직의 understands와 근본적인 의미가 다르다. ASVO로직에서는 다음과 같은 추론은 성립하지 않는다.

(X) P understands $X \Rightarrow P$ has X



〔그림 2〕 상태 전이



〔그림 3〕 전역 상태 전이

Ⅳ. 논리 의미 구조(logic semantics)

ASVO로직의 논리적 의미 구조는 SVO로직을 바탕으로, modal로직의 기본적인 의미 구조 모델인 Kripke의 possible world[15] 구조에서 검증하였다. 다만, 구문 구조의 차이로 인하여, 의미 구조 역시 SVO로직과는 차이가 있다. 먼저 계산 모델과 논리식의 진리 조건을 정의하고 로직의 안정성(soundness)을 증명하도록 한다.

4.1 계산 모델 (model of computation)

계산 모델에 대해서 간략히 설명하면, 시스템 내에서의 모든 계산은 유한한 참여 주체들($P_1, \ldots, P_n$)에 의해서만 이루어지도록 해야하며, 시스템 환경을 표현하기위한 환경 주체(P_e)를 통해서 침입자나 반사된 메시지 등을 모델링할 수 있도록 해야한다.

4.1.1 주체와 상태 전이

각 주체 P_i는 국소 상태(local state) s_i를 가져야 하며, 따라서 전역 상태(global state)는 $n+1$개의 국소 상태들의 집합이라고 할 수 있다. 특히 각 주체는 다음과 같은 세가지 동작(action)을 시스템 내에서 수행할 수 있다. 이것은 SVO로직의 모델에서 기인한다.[11,12]

- 메시지 전송 : *send*(X, G)
- 메시지 수신 : *receive*$()$
- 새로운 데이터 생성 : *generate*(X)

각 동작은 [그림 2]와 같이 국소 상태를 다음 국소 상태로 전이시키며, 특히 메시지 수신의 경우 해당 주체 P_i의 국소 버퍼에 존재하는 수신 메시지에 의존하여 전이하게된다. 따라서 전이 후에 비로소 *receive*(X)에 대해서 명확히 할 수 있다. 또한 *generate*(X)의 경우 반드시 T에 속한 항에 대해서만 수행 가능하다.

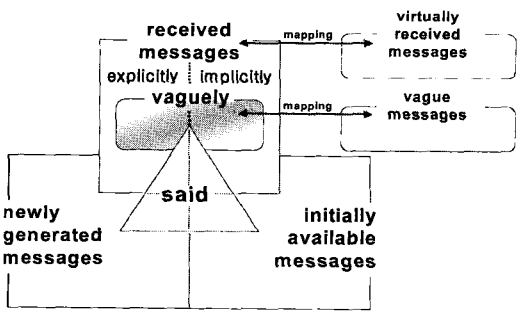
4.1.2 실행 (run) 모델

실행 r은 정수 시간 t로 규정되는 전역 상태들의 무한 열이라고할 수 있으며, 어떤 실행 r에 대한 최초 시간 t_r은 해당 프로토콜 수행 이전의 것들까지 감안하여, 조건 $t_r \leq 0$을 반드시 만족해야한다. 해당 프로토콜 수행을 위한 개시 시간은 반드시 $t=0$으로 정의한다. 한편, 실행 r에서 어떤 시간 t의 전역 상태를 시점 (r, t)로 나타내며, 이것을 다른 말로 possible world라고 한다. [그림 3]은 전역 상태의 전이를 나타낸다. 예를 들면, 실행 r이 수행중인 어떤 시간 t 라고 하면, 이때 주체 P_i가 갖는 국소 상태 s_i를 (r_i, t)로 나타낼 수 있다. 각 국소 상태 s_i에는 (r_i, t) 시점에 이르기까지 주체 P_i가 수행한 동작들에 대한 국소 히스토리와 함께 언어 M_T에 정의된 메시지 구성 규칙의 가용 변환 집합인 A_i가 포함된다. 변환 A_i가 가용하다함은, 주체 P_i가 갖는 계산 능력에 따라서 규정된다는 것을 의미한다. 한편, 환경 주체의 상태, 즉 환경 상태에는 전역 히스토리와 함께 해당 환경에서 가능한 변환들의 집합, 그리고 주체 P_i에게 전송된 메시지에 대한 버퍼 m_i들이 포함된다.

4.1.3 메시지 집합

주체 P_i는 언어 M_T에 정의된 메시지 구성 규칙에 따라 자신에게 가용한 변환 A_i를 통하여 메시지를 구성할 수 있다. 여기서 주체 P_i가 시점 (r_i, t)에서 소유하는 메시지의 집합을 seen messages 집합이라고 하며, $\Sigma(r_i, t)$로 표기한다. 이것은 다음과 같은 부분 집합으로 이루어진다.

- received messages $\Omega(r_i, t)$
- explicitly received messages $\chi(r_i, t)$
- implicitly received messages $\mu(r_i, t)$
- vaguely received messages $\upsilon(r_i, t)$
- newly generated messages $\Psi(r_i, t)$
- initially available messages $\Phi(r_i, t)$
- said messages $\Xi(r_i, t)$
- said submessages $\xi(r_i, t)$



[그림 4] 메시지 집합

특히, seen messages 집합 $\Sigma(r_i, t)$에 포함되지 않는, 즉 주체 P_i에 의해서 직접적으로 소유될 수 없는 유도 메시지의 집합을 virtually received messages $\Pi(r_i, t)$와 vague $\Lambda(r_i, t)$라고 별도로 정의한다. [그림 4]에서는 이와 같은 메시지 집합에 대해서 도시하고 있다. 명심해야할 것은, 여기서 said message란 해당 주체 P_i의 메시지를 의미하는 것이며, 이것은 다른 주체 곧, 해당 메시지를 수신하는 주체에게는 received message 집합에 나타나게 된다.

(가) received messages $\Omega(r_i, t)$

주체 P_i가 시점 (r_i, t)까지 수신한 메시지들에 대한 집합이며, explicitly received messages, implicitly received messages, vaguely received messages 등의 부분 집합으로 구성된다.

1) explicitly received messages $\chi(r_i, t)$

다음과 같이 명백히 수신된 메시지들로 구성된다.

가) 시점 t 또는 그 이전의 시점에서, 국소 메시지 히스토리에 *receive*(X)의 형태로 나타나는 모든 수신 메시지 X.
나) 위의 가)에서 비롯된 메시지 X들의 연결 메시지.
다) X_K가 수신 메시지이며, K에 대한 역변환이 가용한 경우, 주체 P_i가 명확히 이해[6]할 수 있는 메시지 X.
라) 서명된 메시지 X.

2) implicitly received messages $\mu(r_i, t)$

위의 1)의 메시지들로부터, 문맥적으로 명확히 유도된 메시지들로 구성된다.

3) vaguely received messages $\upsilon(r_i, t)$

위의 1)의 다)와 같이 변환된 메시지 중에서 주체 P_i가 명확히 분간할 수 없는 메시지들로 구성된다. 즉, X_K가 수신 메시지이며, K에 대한 역변환이 가용한 경우, 주체 P_i가 명확히 분간할 수 없는 메시지 X의 유도 메시지[7]로 구성된다. 이때 X는 vague messages 집합에 소속되며, 유도 메시지 vague(X)와 서로 사상(mapping) 관계에 있다고 말한다. 만약 vague(X)가 "문맥상" X와 동치인 사실이 확인될 경우, vague(X)는 X로 치환된다.

(나) newly generated messages $\Psi(r_i, t)$

주체 P_i가 시점 (r_i, t)까지 생성한 메시지들의 집합으로서, 시점 t 또는 그 이전의 시점에서, 국소 메시지 히스토리에 *generate*(X)의 형태로 나타나는 모든 생성 메시지 X로 구성된다.

(다) initially available messages $\Phi(r_i, t)$

주체 P_i에게 시점 $(r_i, 0)$에 최초로 주어진 메시지들의 집합이다.

(라) said messages $\Xi(r_i, t)$

주체 P_i가 시점 (r_i, t)에 명확히 송신한 메시지들의 집합으로서, 시점 (r_i, t)에 주어진 seen messages 집합에서 도출된다. 즉, 만약 *send*(X, G)가 시점 (r_i, t)의 국소 히스토리에 나타난다면, 이것은 시점 (r_i, t)에서 메시지 X가 seen messages 집합의 원소임을 의미한다. 하지만 해당 메시지로부터 임의로 유도될 수 있는 메시지나 수신자에 의해서 분간되지 않은 상태의 불명확한[8] 메시지들까지 항상 포함하지는 않는다. 한편, said messages 집합은 같은 시점에서의 여러 유형의 said submessage들로 구성된다.

1) said submessages $\xi(r_i, t)$

주체 P_i가 시점 (r_i, t)에 송신한 메시지 M과 다음과 같은 명확한 메시지들로 구성된다. 단, 모든 submessage들은 반드시 주체 P_i가 시점 (r_i, t)에 소유하고 있는 메시지이다.

가) 메시지 M의 submessage들의 연결 메시지.
나) 메시지 M의 암호화된 submessage들 중에서 주체 P_i가 암호키를 가지고 직접 암호화한 경우의 복호 메시지, 즉 해당 암호 블록의 submessage.
다) 메시지 M의 서명된 submessage들 중에서 주체 P_i가 직접 서명한 경우의 비서명된 submessage.
라) 메시지 M의 해쉬값 submessage들 중에서 주체

6) 논리식 P_i understands X에 해당한다.
7) 논리식 vague(X)로 나타낸다.
8) 구체적으로 어떤 메시지 X가 수신자에 의해서 vague(X) 혹은 vague(X')로 해석되는 경우를 불명확하다고 일컫는다.

P_i가 직접 해쉬값을 구한 경우의 원본 메시지.

마) 메시지 M의 submessage를 통해서 주체 P_i가 전달하려고 의도하는 구체적인 유도 메시지 M'[9].

(마) virtually received messages $\Pi(r_i, t)$

주체 P_i의 seen messages (r_i, t) 집합의 어떤 원소와 사상 관계를 갖는 메시지로 구성된다. 이것은 seen messages (r_i, t) 집합에 직접 포함되지는 않으며 다만 사상 관계만을 가질뿐이다. 사상 관계의 의미는 $t' < t$일 경우, 만약 $send(X, G)$가 어떤 주체 Q_j의 시점 (r_j, t')의 국소 히스토리에 나타나며, $receive(X')$이 P_i의 시점 (r_i, t)의 국소 히스토리에 나타난다면, 해당 메시지를 수신한 주체 P_i가 이에 대해서 해석할 수 있는지를 표현하는 것이다. 따라서 무사히 해석될 경우에는 P_i에 의해서 said message나 received message로 유도된다. 특히 메시지의 일부는 vague message (r_i, t) 집합의 원소이며, 이 경우에는 주체 P_i에 의해서 식별되는 경우 seen messages (r_i, t) 집합의 사상 메시지와 직접 치환될 수 있다. 이것은 대칭키를 사용하는 경우, 메시지의 근원지 판별을 위해서 사용하거나, 일방향 함수를 사용할 경우 입력값에 대한 식별을 위해서 사용될 수 있다.

가) 메시지 X의 submessage들의 연결 메시지.

나) 메시지 X의 암호화된 submessage들 중에서 주체 Q_j가 암호키를 가지고 직접 암호화한 경우의 복호 메시지, 즉 해당 암호 블록의 submessage.

다) 메시지 X의 해쉬값 submessage들 중에서 주체 Q_j가 직접 해쉬값을 구한 경우의 원본 메시지.

라) 메시지 X의 submessage를 통해서 주체 Q_j가 전달하려고 의도하는 구체적인 유도 메시지 X'[10].

(바) vague messages $\Lambda(r_i, t)$

주체 P_i의 received messages (r_i, t) 집합의 부분집합인 vaguely received messages를 유도하는 메시지들로 구성되었으며, 각 메시지는 해당 유도 메시지와 서로 사상 관계를 갖는다. 이것은 seen messages (r_i, t) 집합에 포함되지 않지만, 주체에 의해서 식별되는 경우, 사상 메시지와 치환될 될 수 있다. 즉, 이것은 메시지 식별을 위해서 사용된다.

4.2 Modal 연산

ASVO로직은 신뢰 기반의 modal로직, 즉 doxastic 로직이다. 따라서 ASVO로직의 modal 연산자는 believes predicate에 해당한다. ASVO로직은 SVO로직과 마찬가지로, possible world에 근간하여, 본 로직의 modal 연산자에 해당하는 신뢰에 대해서 그 의미구조를 정의한다. 다시 말하면, 주어진 상태에서 주체의 신뢰란, 어떤 world들이 해당 주체의 상태로서 가능할 수 있느냐에 따라서 결정된다고 할 수 있다. 하지만 주체의 관점에서 이러한 world들을 서로 구별할 수 없으므로, 각 주체 P_i에 대해서 이것을 지시해줄 수 있는 관계 (relation)를 정의할 필요가 있다.

관계 부호 $\hookrightarrow_i$는 실제 world (r, t)마다 어떤 world들이 P_i를 위해서 가능한지를 나타낸다. 즉, 이것은 possible world에 대한 추이적 접근성 관계(transitive accessibility relation)를 나타내며, 물론 P_i에 의해서 이해되고 식별될 수 있는 메시지와 명백히 관련이 있다. 우선 world (r, t)에서 주체 P_i에게 주어지는 모든 메시지는 seen messages (r, t) 집합과 그 외의 유효 집합에 나타나야한다. 여기서 seen messages (r, t) 집합에 존재하는 메시지들은 P_i에 의해서 소유[11]될 수 있는 메시지인 반면, 그 외의 집합에 해당하는 virtually received messages (r, t) 집합이나 vague messages (r, t) 집합에 존재하는 메시지들은 P_i에 의해서 직접적으로 소유될 수 없는 메시지이다. 따라서 P_i에 의해서 이해되고 식별될 수 있는 메시지는 seen messages (r, t) 집합에 존재하는 메시지들이라고 할 수 있다. 하지만 virtually received messages (r, t) 집합이나 vague messages (r, t) 집합에 존재하는 메시지들은 seen messages (r, t) 집합의 특정 메시지들과 사상 (mapping) 관계를 갖게 되며, 결국 이들은 관계된 world에서 이해 (understanding)에 관한 의미적 조건에 따라서 서로 일정하게 치환될 수 있는 동치 관계를 갖는다.

따라서, 실제 world에 존재하는 메시지인 seen messages (r, t) 집합의 메시지와, 이에 대해 사상 가능한 메시지인 virtually received messages (r, t) 집합이나 vague messages (r, t) 집합의 메시지들 모두 world의 관계 $\hookrightarrow_i$와 관련이 있다.

9) 예를 들면, $P \xleftrightarrow{K} Q$.

10) 예를 들면, X from Q.

11) 논리식 P has X로 나타낼 수 있다.

$(r,t) \hookrightarrow_i (r',t')$의 의미는 주체 P_i가 world (r,t)에서 어떤 신뢰를 갖는다면, world (r',t')는 해당 신뢰를 유지할 수 있는 possible world라는 것이다. 즉, 이것은 일종의 possibility 관계라고 할 수 있으며, world (r,t)와 (r',t')에서 메시지 집합과 그 구성 요소들이 의미구조적으로 동일하게 유지되어야한다.

4.3 진리 조건(truth conditions)

먼저 계산 모델에 의해서 표현되는 시스템을 실행들의 집합 R이라고 규정하고, 논리식이 참인 진리값을 가질 조건에 대해서 기술한다. 어떤 시점 (r,t)에서 논리식 φ가 참인 경우 다음과 같이 표기한다.

$(r,t) \models \varphi$

그리고 $\models \varphi$는 논리식 φ가 시점에 상관없이 항상 참이라는 사실을 나타낸다. 또한 if$f$는 if and only if를 의미한다.

T1) 논리 연결(logical connectives)

$$(r,t) \models \varphi \wedge \psi$$

if$f$ $(r,t) \models \varphi$이며 $(r,t) \models \psi$이다.

T2) 송신(saying)

$$(r,t) \models P \text{ said } X$$

if$f$ 실행 r에서 $t' \le t$를 만족하는 어떤 시점에 P가 메시지 M을 송신했으며, 여기서 X는 (r,t')에서 P의 메시지 M의 said submessage이다.

$$(r,t) \models P \text{ says } X$$

if$f$ 실행 r에서 $0 \le t' \le t$를 만족하는 어떤 시점에 P가 메시지 M을 송신했으며, 여기서 X는 (r,t')에서 P의 메시지 M의 said submessage이다.

T3) 수신(receiving)

$$(r,t) \models P \text{ received } X$$

if$f$ X는 (r,t)에서 P의 received message 집합의 원소이다.[12]

$$(r,t) \models X \text{ from } Q$$

if$f$ X는 (r,t)에서 received message 집합의 임의의 원소와 동치 사상 관계를 갖는 메시지이며, X from Q는 (r,t)에서 P의 virtually received message 집합의 원소이다.

$$(r,t) \models P \text{ received2 } X$$

if$f$ X는 (r,t)에서 P의 received message 집합의 원소와 사상 관계를 갖는 virtually received message 집합의 원소의 연결 혹은 결합이다.

T4) 소유(having)

$$(r,t) \models P \text{ has } X$$

if$f$ X는 (r,t)에서 P의 seen message 집합의 원소이다.

T5) 이해(understanding)

$$(r,t) \models P \text{ understands } X$$

if$f$ X는 (r,t)에서 P의 seen message 집합의 원소이거나, seen message 집합의 원소와 vague message 집합의 원소의 연결 또는 결합이다.

$$(r,t) \models \text{vague}(X)$$

if$f$ 모든 시점 t'에서,

(1) X는 (r,t')에서 P의 vague message 집합의 원소이며, 이와 사상 관계를 갖는 vague(X)는 vaguely received message 집합의 원소이다.

(2) $(r,t') \models P$ understands X이면, vague(X)는 X로 치환된다.

T6) 관할(jurisdiction)

$$(r,t) \models P \text{ controls } \varphi$$

if$f$ $t' \ge 0$을 만족하는 모든 시점에서,

(1) $(r,t') \models P$ says φ가 $(r,t') \models \varphi$를 유도한다.

(2) φ의 목적이 되는 메시지 X에 대해서 $(r,t') \models P$ says X가 $(r,t') \models P$ says φ를 유도한다.

T7) 신규(freshness)

$$(r,t) \models \text{fresh}(X)$$

12) 이때 X가 vaguely received message가 아니면, $(r',t') \models R$ said X이고 $t' < t$이다.

if$f$ 모든 시간 $t'<0$에 모든 주체 P에 대해서 ¬ $((r,t') \models P$ said $\varphi)$이다.

T8) 키(keys)

$$(r,t) \models P \overset{K}{\Longleftrightarrow} Q$$

if$f$ 모든 시점 t'에서,

(1) $(r,t') \models R$ said X_K이면, $(r,t') \models R$ received X_K이거나, 또는 $R \in P,Q$이며 $(r,t') \models R$ said X이며 $(r,t') \models R$ has K이다.

(2) $(r,t') \models R$ received X_K이면, $(r,t') \models R$ received2 X from S이며 $R,S \in P,Q$이다.

(3) $(r,t') \models R$ received2 X from $Q \wedge (r,t') \models R$ understands X이면, $R=P$이며 $(r,t') \models Q$ said X이며 $(r,t') \models Q$ said $R \overset{K}{\Longleftrightarrow} Q$이며 $(r,t') \models Q$ has K이다.

(4) $(r,t') \models$ fresh(K)이면, $(r,t') \models$ fresh($R \overset{K}{\Longleftrightarrow} S$)이며 $R,S \in P,Q$이다.

(5) $(r,t') \models R$ has K이며 $R \in P,Q$이다.

전자서명 메시지를 수신하는 경우, 송신자의 공개키에 대한 검증이 되면, 해당 메시지와 근원지의 무결성을 확인할 수 있다.

$$(r,t) \models \mathrm{PK}_\sigma(P,K)$$

if$f$ 모든 시점 t'에서,

(1) $(r,t') \models Q$ received $X_{K^{-1}}$이면, $(r,t') \models P$ said X이며 $(r,t') \models P$ received X다.

(2) $(r,t') \models P$ has K^{-1}이다.

$$(r,t) \models \mathrm{PK}_\psi(P,K)$$

if$f$ 모든 시점 t'에서,

(1) $(r,t') \models Q$ received X_K이면, $(r,t') \models Q$ received vague(X)이며 $Q=P$이다.

(2) $(r,t') \models Q$ received $X_K \wedge (r,t') \models Q$ understands X이면, $(r,t') \models Q$ received X이며 $Q=P$이다.

(3) $(r,t') \models P$ has K^{-1}이다.

$$(r,t) \models \mathrm{PK}_\delta(P,K)$$

if$f$ 모든 시점 t'에서,

(1) 어떤 Q와 K_q에 대해서, $(r,t') \models Q$ says K_q이고 ¬$((r,t') \models \mathrm{PK}_\delta(R,K))$이면, $(r,t') \models P \overset{F_0}{\Longleftrightarrow} (K,K_q)Q$이다.

(2) 어떤 K_q에 대해서, $(r,t') \models P$ has K_q이고 ¬$((r,t') \models \mathrm{PK}_\delta(R,K))$이면, $(r,t') \models P$ has $F_0(K_p,K_q)$이다.

(3) 모든 R과 K_r에 대해서, ¬$((r,t') \models R \overset{F_0(K_r,K)}{\Longleftrightarrow} P)$이면, 모든 S와 K_s에 대해서 ¬$((r,t') \models R \overset{F_0}{\Longleftrightarrow} (K_r,K_s)S)$이다.

(4) $(r,t') \models P$ has $F_0(K_p,K_q) \equiv P$ has K_{pq}이다.

T9) 신뢰 (believing)

$$(r,t) \models P_i \text{ believes } \varphi$$

if$f$ $(r,t) \hookrightarrow_i (r',t')$인 모든 world (r',t')에서 $(r',t') \models \varphi(r',t')$이며, 어떤 world (r',t')에서 $\varphi = \varphi(r',t')$이다.

4.4 로직의 안정성

$\Gamma \models \varphi$란 일련의 논리식 집합 Γ가 모두 진리값 참을 갖는 world에서는 논리식 φ도 진리값 참을 갖는다는 의미이다. 또한 $\Gamma \vdash \varphi$란 일련의 논리식 집합 Γ로부터 논리식 φ를 유도할 수 있다는 의미이다. 따라서 다음은 로직의 안정성에 관한 정리이다.

[정리 1] (안정성: Soundness)

ASVO로직은 안정성을 갖는다: $\Gamma \vdash \varphi$이면, $\Gamma \models \varphi$이다.

(증명)

주체 P_i에게 world (r,t)에서 주어지는 메시지의 집합은 $\Sigma = \Omega, \Psi, \Phi, \Xi$, $\Omega = \chi, \mu, \upsilon$, $\Xi = \xi$, Π, Λ이다. 여기서 $\Omega(r,t)$와 $\Pi(r,t)$는 서로 사상 관계를 가지며, $\upsilon(r,t)$와 $\Lambda(r,t)$는 서로 사상 관계를 갖는다. 하지만 $\Sigma(r,t)$는 P_i에 의해서 소유될 수 있는 반면, $\Pi(r,t)$와 $\Lambda(r,t)$는 직접 소유될 수 없다. 다만 논리적으로 이해(understanding)에 관한 조건을 만족할 때만 치환할 수 있다. $(r,t) \hookrightarrow_i (r',t')$는 이와 같은 사상 관계하에서 동치 관계를 나타낸다.

A1~A4) ASVO로직의 신뢰 공리 A1, A2, A3, A4는 modal로직의 기본 공리인 K, T, 4, 5에 해당한다. 또한 world (r, t)에 대한 관계 $\hookrightarrow_i$는 possible world의 동치 관계를 의미하므로 modal로직의 기본 공리인 이들은 모두 성립한다.

A5) P understands $X \wedge P$ received2 X from $Q \supset Q$ said $(X, P \xleftrightarrow{K} Q)$

만약 $(r, t) \models P$ received2 X from Q라면, 수신의 received2에 관한 진리 조건과 from에 관한 진리 조건에 의하여, world (r, t)에서 $(X$ from $Q) \in \Pi$가 존재하며, 또한 X에 대한 사상 관계를 갖는 $x \in \Omega$가 존재한다. 여기서 Π의 성질에 따라서 $X \in \Lambda$일 수 있으며, 이때 $(r, t) \models P$ understands X라면, 이해에 관한 진리 조건에 의하여, Ω의 x를 X로 치환할 수 있다. 즉, P는 X에 대한 사실을 신뢰할 수 있다. 따라서 만약 $(r, t') \models R$ received2 X from $Q \wedge (r, t') \models R$ understands X이면, $P \xleftrightarrow{K} Q$에 관한 진리 조건에 의하여, 모든 시점 t'에서 $R = P$이며 $(r, t') \models Q$said X이며 $(r, t') \models Q$ said $R \xleftrightarrow{K} Q$이며 $(r, t') \models Q$ has K이다. 곧 Π에 대한 정의에 의하여, $t'' < t$일때 $(r, t'') \models Q$ said X이며 $(r, t'') \models Q$ said $P \xleftrightarrow{K} Q$이다. 곧 $(r, t'') \models Q$ said $(X, P \xleftrightarrow{K} Q)$이다.

A5a1) $(\mathrm{PK}_\sigma(\mathrm{Q}, \mathrm{K}) \wedge P$ received $\{X\}_{K^{-1}}) \supset Q$ said $X \wedge P$ received X

만약 $(r, t) \models \mathrm{PK}_\sigma(Q, K)$이라면, $\mathrm{PK}_\sigma(P, K)$에 관한 진리 조건에 의하여, 모든 시점 t'에서 P received $\{X\}_{K^{-1}}$인 경우 $(r, t') \models P$ said X이며 $(r, t') \models P$ received X이다.

A6) $(\mathrm{PK}_\delta(\mathrm{P}, \mathrm{K}_\mathrm{p}) \wedge \mathrm{PK}_\delta(\mathrm{Q}, \mathrm{K}_\mathrm{q})) \supset P \xleftrightarrow{K_{pq}} Q$

만약 $(r, t) \models \mathrm{PK}_\delta(\mathrm{P}, \mathrm{K}_\mathrm{p})$이고 $(r, t) \models \mathrm{PK}_\delta(Q, K_q))$이지만, $\neg((r, t) \models P \xleftrightarrow{K_{pq}} Q)$이라고 가정하면, $\mathrm{PK}_\delta(P, K)$에 관한 진리 조건에 의하여, 모든 시점 t'에서, 모든 R과 K_r에 대해서 $(r, t') \models Q$ says K_q이면 항상 $\neg((r, t') \models Q \xleftrightarrow{K_{qr}} R)$이어야 한다. 하지만 이것은 초기 가정 $(r, t) \models \mathrm{PK}_\delta(\mathrm{Q}, \mathrm{K}_\mathrm{q})$에 위배된다. 따라서 만약 $(r, t) \models \mathrm{PK}_\delta(\mathrm{P}, \mathrm{K}_\mathrm{p})$이고 $(r, t) \models \mathrm{PK}_\delta(Q, K_q))$이면, $(r, t) \models P \xleftrightarrow{K_{pq}} Q$이다.

A6a1~A7a1) $\mathrm{PK}_\delta(P, K)$에 관한 진리 조건에 의하여 명백히 성립한다.

A8~10) 수신에 관한 진리 조건에 의하여 명백히 성립한다. A8)은 수신된 메시지의 submessage 구성에 해당하며, A9)와 A9a1)은 수신된 메시지가 $\{X\}_K$일 경우, X를 식별할 수 없을 경우 $X \in \Lambda$이며, 식별 가능할 경우 $X \in \Omega$이라는 진리 조건의 성립을 의미한다.

A9a2) P received vague$(\{X\}_K) \supset P$ received {vague $(X)\}_K$

만약 $(r, t) \models P$ has K^{-1}라고 하면 A9)에 의하여,

(1) $(r, t) \models P$ received vague$(\{X\}_K)$이면, vague$(\{X\}_K) \in \Omega$와 $\{X\}_K \in \Lambda$는 서로 사상 관계를 가지며, $(r, t) \models P$ understands $\{X\}_K$일 경우 $\{X\}_K \in \Omega$로 치환된다. 즉, $(r, t) \models P$ received $\{X\}_K$이므로, vague$(X) \in \Omega$와 $X \in \Lambda$는 서로 사상 관계를 가지며, $(r, t) \models P$ understands X일 경우 $X \in \Omega$로 치환된다. 즉, $(r, t) \models P$ understands $\{X\}_K$이고 $(r, t) \models P$ understands X일 경우, $(r, t) \models P$ received vague$(\{X\}_K)$이면, $(r, t) \models P$ received X이다.

(2) $(r, t) \models P$ received {vague$(X)\}_K$이면, vague(vague $(X)) \in \Omega$와 vague$(X) \in \Lambda$는 서로 사상 관계를 가지며, $(r, t) \models P$ understands vague(X)일 경우 vague$(X) \in \Omega$와 $X \in \Lambda$로 치환된다. 즉, $(r, t) \models P$ received vague(X)이므로, vague$(X) \in \Omega$와 $X \in \Lambda$는 서로 사상 관계를 가지며, $(r, t) \models P$ understands X일 경우 $X \in \Omega$로 치환된다. 즉, $(r, t) \models P$ understands vague(X)이고 $(r, t) \models P$ understands X일 경우, $(r, t) \models P$ received {vague $(X)\}_K$이면, $(r, t) \models P$ received X이다.

(3) $(r, t) \models P$ understands $\{X\}_K$이면 $\neg((r, t) \models P$ understands vague$(X))$라고 할때, A9)에 의해서 $(r, t) \models P$ received $\{X\}_K$이면 $(r, t) \models P$ received vague(X)이다. 즉, 이해의 진리 조건에 의해서 $(r, t) \models P$ understands vague(X)이므로 모순이며, 따라서 $(r, t) \models P$ understands $\{X\}_K$이면$((r, t) \models P$ understands vague(X)이다.

결과적으로 $(r, t) \models P$ received vague$(\{X\}_K)$이면 $(r, t) \models P$ received {vague$(X)\}_K$이다.

A10~A14) 소유에 관한 진리 조건에 의하여 명백히 성립한다.

A15~A16a) 키와 소유에 관한 진리 조건에 의하여 명백히 성립한다.
A17~A17a1) 송신에 관한 진리 조건에 의하여 명백히 성립한다.
A18~A19) 신규에 관한 진리 조건에 의하여 명백히 성립한다.
A20) 관할에 관한 진리 조건에 의하여 명백히 성립한다.
A21) 송신과 신규에 관한 진리 조건에 의하여 명백히 성립한다.
A22~A22a1) 키에 관한 진리 조건에 의하여 명백히 성립한다.
A23~A23a1) 수신의 from과 received2에 관한 진리 조건에 의하여 명백히 성립한다.
A24~A24a1) 관할에 관한 진리 조건 (2)에 의하여 명백히 성립한다.
A25) 신규에 관한 진리 조건에 의하여 명백히 성립한다.
A26~27) 소유에 관한 진리 조건에 의하여 명백히 성립한다.
A28~A28a1) 소유와 이해에 관한 진리 조건에 의하여 명백히 성립한다.

Γ로부터 유도된 논리식 φ은 자체가 위의 공리를 바탕으로 구성된 정리이거나, MP 혹은 Nec을 통해서 유도된 경우이다. 따라서 이와 같은 경우는 $\Gamma \vdash \varphi$가 명백히 성립한다.

V. Isabelle/ASVO를 이용한 검증 방법

5.1 ASVO 로직의 증명 목표(Goals)

ASVO로직을 통하여 암호 프로토콜의 안전성에 대한 증명을 하기 위해서는, 프로토콜이 정확히 만족해야할 목표를 설정하는 것이 중요하다. 인증 및 키 분배를 위한 암호 프로토콜이 이루어야할 목표에 대해서는 BAN로직과 SVO로직에서 각각 정형화하였다.[2,11,12] ASVO 로직은 이를 바탕으로 하여 논리 검증 목표를 다음과 같이 규정한다. 기존 로직의 목표와 유사한 면이 있지만 predicate의 의미에 따라서 명확히 차이가 있다. 증명 목표에 대하여 보다 더 업데이트된 사항은 참고문헌 [17]에서 찾을 수 있다.

5.1.1 프로토콜 활성화(Activeness)

온라인 상태인 어떤 상대방과 프로토콜을 수행하였음을 확인하기 위해서는, 기본적으로 원하는 메시지, 예를 들면 X를 수신했는지 여부를 확인할 수 있다. 하지만 이 경우 상대방에 대한 신원 확인은 요구하지 않으며, 따라서 다음과 같은 논리식으로 나타낼 수 있다.

ASVO0 : A believes A received X

그러나 이것은 메시지 X에 대한 신규성 확인을 포함하지 않는다. 따라서 프로토콜의 현재성을 함께 확인하기 위해서는 다음과 같은 논리식을 만족해야 한다.

ASVO0 : A believes A received $X \wedge A$ believes fresh (X)

5.1.2 온라인 상태(Aliveness)

현재 온라인 상태인 특정 상대방과 프로토콜을 수행하고 있는지 확인하기 위해서, 즉 상대방의 신원을

〔표 1〕 ASVO 로직 목표

ASVO 목표	논리식
ASVO0: 활성화 (Activeness)	A believes A recieved X
	A believes A recieved $X \wedge A$ believes fresh(X)
ASVO1: 온라인 상태 (Aliveness)	A believes B says X
ASVO2: 인증 (Authentication)	A believes B says $F(X) \wedge A$ believes A has X
	A believes B says $A \overset{K}{\leftrightarrow} B \wedge A$ believes $A \overset{K}{\leftrightarrow} B$
ASVO3: 비검증된 키 분배	A believes A has $K \wedge A$ believes fresh(K)
ASVO4: 검증된 키 분배	A believes $A \overset{K}{\leftrightarrow} B \wedge A$ believes fresh(K)
ASVO5: 키 신규성 (Key freshness)	A believes fresh(K)
ASVO6: 키 분배 상호 확인	A believes B says $A \overset{K}{\leftrightarrow} B$

묵시적으로 확인하기 위해서는, 프로토콜 참여자가 상대방의 송신과 시점에 대한 신뢰를 얻을 수 있어야한다. 이것을 논리식으로 나타내면 다음과 같다.

ASVO1 : A believes B says X

의미 구조적으로 A가 B와의 프로토콜 수행 사실을 확인한다는 면에서 묵시적으로 상대방을 인증하게된다.

5.1.3 인증(Authentication)

상대방에 대한 인증을 위해서는 온라인 상태에 대한 묵시적 인증과 비교할 때 보다 명확한 수행 사실이 추가되어야한다. 즉, 프로토콜 참여자가 송신한 challenge에 대해서 의존적인 응답 사실을 상대방으로부터 확인할 수 있어야한다. 따라서 다음과 같은 논리식을 얻을 수 있다.

ASVO2 : A believes B says $F(X) \wedge A$ believes A has X

하지만 이와 같은 경우, 상대방이 자신과의 통신에 대한 신뢰를 하고 있는지 확인할 수 없다. 즉, 상호 인증에 관련된 사실을 확인할 수 없다. 따라서 키 분배를 통해서 상대방에 대한 인증과 함께, 상대방의 인증 여부를 확인할 수 있는 논리식을 다음과 같이 얻을 수 있다.

ASVO2' : A believes B says $A \overset{K}{\longleftrightarrow} B \wedge A$ believes $A \overset{K}{\longleftrightarrow} B$

즉, 참여자 A는 B와 서로 같은 키를 공유하게된 사실을 신뢰하며, B 또한 이에 대한 사실을 인정하

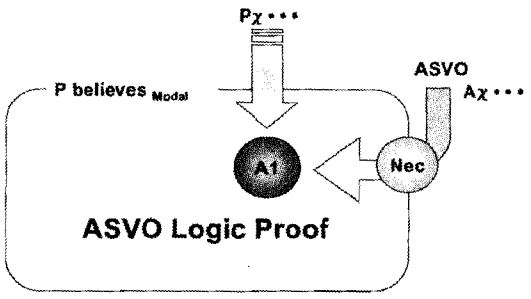


그림 5) $\vdash P$ believes $\varphi \Rightarrow P$ believes ϕ

고 있음을 확인할 수 있다.

5.1.4 비검증된 키 분배

키 분배를 위해서는, 키 유형의 값 K에 대한 소유 사실과 해당 키의 신규성에 대한 확인이 필요하다. 하지만 이것은 비검증된 키 분배라고 할 수 있는데, 그 의미는 해당 키를 이용해서 통신할 상대방에 대해서 명확히 규정되지 않았다는 것이다. 비검증된 키 분배에 대한 확인을 위한 논리식은 다음과 같다.

ASVO3 : A believes A has $K \wedge A$ believes fresh (K)

5.1.5 검증된 키 분배

검증된 키 분배를 위해서는, 키를 상대방과 공유한다는 사실과 해당 키의 신규성에 대한 확인이 필요하다. 검증된 키 분배에 대한 확인을 위한 논리식은 다음과 같다.

ASVO4 : A believes $A \overset{K}{\longleftrightarrow} B \wedge A$ believes fresh(K)

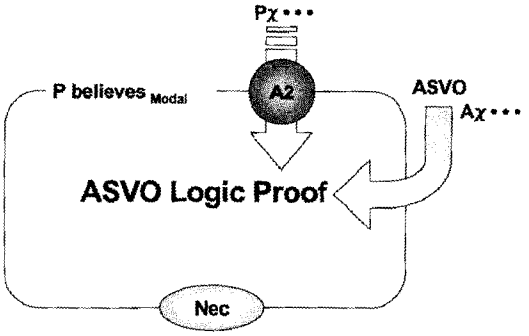


그림 6) $\vdash P$ believes ($\varphi \Rightarrow \phi$)

5.1.6 키 신규성 (Key freshness)

위에서 이미 언급되었던 키의 신규성에 대한 확인을 위한 논리식을 분리하면 다음과 같다.

ASVO5 : A believes fresh(K)

5.1.7 키 분배 상호 확인

위에서 언급되었던 키 분배에 대해서 상호 확' 하도록 하기 위한 논리식을 분리하면 다음과 같

ASVO6 : A believes B says $A \overset{K}{\longleftrightarrow} B$

5.2 ASVO 로직을 통한 검증 방법

5.2.1 기본 개념

ASVO 로직은 프로토콜 증명 단계를 간소화하고, 자동 검증을 지원하여 정확한 추론 경로를 쉽게 찾도록 설계되었다. 따라서 Isabelle/Isar로 구현된 Isabelle /ASVO 시스템을 통해서 암호 프로토콜의 인증 및 키 분배에 관한 반자동 검증을 지원할 뿐만 아니라, SVO 로직의 단순 명료성을 바탕으로 설계된 만큼 전문가에 의한 수동 증명도 가능하다. Isabelle/Isar를 선택한 이유는 구현이 용이한 해석환경을 제공하며, ML 언어보다 쉬운 Isar 언어를 사용할 수 있기 때문이다. Isabelle/ASVO 시스템을 통한 증명 과정은 후향 증명으로 이루어진다.[10] ASVO 로직은 다음과 같이 명확한 가정으로부터 일정의 프로토콜 목표를 유도할 수 있는지, 논리적으로 검증하기 위한 도구라고 할 수 있다.

$$\vdash \text{가정1; 가정2; } \cdots \text{, 가정 } n \Rightarrow \text{목표}$$

ASVO 로직을 통하여 프로토콜을 검증하는 절차는 다음과 같다.

① 프로토콜을 기술한다.
② 프로토콜이 달성하고자 하는 목표를 기술한다.
③ 프로토콜의 초기 상태 가정(ISA: initial state assumption)을 기술한다. 이것은 프로토콜의 초기 설정과 관련된 주체의 신뢰를 명시한다.
④ 프로토콜의 수신 메시지 가정(RMA: received message assumption)을 기술한다. 이것은 프로토콜의 정확한 메시지 수신에 대한 주체의 신뢰를 명시한다.
⑤ 로직을 이용하여 프로토콜 참여 주체가 갖는 신뢰를 도출한다.

즉, ASVO 로직의 검증 절차는 기존 SVO 로직의 절차에 비해서 간소화되었음을 알 수 있다. 이미 앞에서 살펴보았듯이 ASVO 로직은 기존의 SVO 로직보다 구체화된 공리 스키마를 갖지만, 의미구조적으로 안정성을 유지한다.

5.2.2 로직 적용 방법

ASVO 로직이 비록 SVO 로직에 바탕을 두고 설계되었지만, 큰 차이점을 다음과 같이 요약할 수 있다.

첫째로 프로토콜의 이상화(idealization) 과정이 매우 간소화되었다. 예를 들면, 기존의 SVO 로직에서는 A received $\{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$와 같은 구문에 대하여, 다음과 같이 이상화하여야했다.

$$A \text{ believes } (A \text{ received } \{N_A, B, *_1.*_2\}_{K_{AS}}$$
$$\Rightarrow A \text{ received } \left\{N_A, B, A \xleftrightarrow{K_{AB}} B, fresh(K_{AB}), *_2\right\}_{K_{AS}})$$

즉, 이미 이상화 과정에 프로토콜 참여자의 이해와 해석에 관하여 수동적인 분석을 해야하며, 향후 분석과정에서도 $*_2$와 같이 미확인된 부분에 대해서 수동적인 복원을 요한다. 하지만 ASVO 로직에서는 이와 같은 이상화 과정을 간소하게 하여 다음과 같이 간략한 가정을 선언할 수 있다.

$$A \text{ believes } A \text{ received } \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$$

둘째로 프로토콜 참여자의 메시지에 대한 이해와 해석을 로직 내부에서 논리적으로 다루었다. 따라서 said, says, received 등과 같은 주요 predicate들의 의미가 SVO 로직의 그것들과는 조금씩 다르다. 예를 들면, 프로토콜 상대방에 대하여 said나 says를 통한 논리식을 도출하기 위해서는, 메시지에 대한 구체적인 이해와 해석이 논리적으로 완성되어야한다. 또한 분명히 해석되지 않은 메시지에 대한 received predicate인 received2와 일반적인 received를 구분하여 논리식이 갖는 수신에 대한 의미를 formal하게 규정하였다. 뿐만 아니라, GNY 로직이나 SVD 로직과 마찬가지로 메시지의 인식성 (recognizability)을 다룰 수 있는 predicate을 규정하였으며, 이것을 understands라고 선언하였다.

셋째로 로직을 normal 로직으로 규정하여, 다음과 같은 Gödel의 두 번째 공리에 바탕을 둔 증명이 가능하도록 하였다.

$$P \text{ believes } (\varphi \rightarrow \psi)$$
$$\Rightarrow (P \text{ believes } \varphi \rightarrow P \text{ believes } \psi)$$

즉, $\vdash P$ believes $\varphi \Rightarrow P$ believes ψ의 증명을 위해서 $\vdash P$ believes $(\varphi \Rightarrow \psi)$에 대한 증명을 하여도 된다고 볼 수 있다. 이와 같은 개념을 그림으로 도시하면 [그림 5] 및 [그림 6]과 같다.

즉, 어떤 증명 과정에서 P believes를 해당 정리에

대한 modal 연산자로 규정하며, 모든 가정 공리들은 $\vdash P$ believes $\varphi \Rightarrow P$ believes ψ와 같은 유형의 정리 증명이나, $\vdash P$ believes ($\varphi \Rightarrow \psi$)와 같은 유형의 정리 증명을 수행할 수 있다.

예를 들어서 다음과 같은 가정과 정리가 주어졌을 경우, ASVO 로직을 이용하여 두 가지 방법으로 같은 결과를 얻을 수 있다.

P1) A believes A received $\{X\}_{K_{AB}}$

P2) A believes $A \xleftrightarrow{K_{AB}} B$

$\vdash A$ believes A received $\{X\}_{K_{AB}} \Rightarrow A$ believes A received2 X from B

먼저 $\vdash P$ believes $\varphi \Rightarrow P$ believes ψ와 같은 유형의 정리 증명을 수행하면 그 절차는 다음과 같다.

P1) A believes A received $\{X\}_{K_{AB}}$

P2) A believes $A \xleftrightarrow{K_{AB}} B$

A23) ($P \xleftrightarrow{K} Q \wedge P$ received $\{X\}_K$)
$\Rightarrow P$ received2 X from Q

Nec) $\vdash \varphi \Rightarrow \vdash P$ believes φ

A1) P believes $\varphi \wedge P$ believes ($\varphi \rightarrow \psi$)
$\Rightarrow P$ believes ψ

MP) φ ; ($\varphi \rightarrow \psi$) $\Rightarrow \psi$

$\Rightarrow A$ believes A received2 X from B

한편 $\vdash P$ believes ($\varphi \Rightarrow \psi$)와 같은 유형의 정리 증명을 수행하면 그 절차는 다음과 같다. 이와 같은 정리 증명은 연역적 행위인 MP 적용을 제외하고는 반드시 후향 증명을 통해서 이루어져야 한다.

P1) A believes A received $\{X\}_{K_{AB}}$

A2) P believes $\varphi \Rightarrow \varphi$

P2) A believes $A \xleftrightarrow{K_{AB}} B$

A2) P believes $\varphi \Rightarrow \varphi$

A23) ($P \xleftrightarrow{K} Q \wedge P$ received $\{X\}_K$)
$\Rightarrow P$ received2 X from Q

Nec) $\vdash \varphi \Rightarrow \vdash P$ believes φ

MP) φ ; ($\varphi \rightarrow \psi$) $\Rightarrow \psi$

$\Rightarrow A$ believes A received2 X from B

이와 같은 증명 과정에서 주목해야할 것은, 후향 증명을 통해서 modal 연산자에 대한 관리가 명확하게 이루어질 수 있다는 것이다. 예를 들어서, 위의 과정의 A received2 X from B에 대해서 해당 world (r, t)에서 얻을 수 있는 신뢰는 주체 A에 대해서 한정된다. 따라서 결과에 Necessitation 규칙을 적용할 경우, B believes A received2 X from B와 같이 임의로 modal 연산자를 적용하지 않는다. 즉, 이 때는 A believes A received2 X from B만 추론 가능하다.

5.3 소프트웨어를 이용한 증명 방법

ASVO 로직의 반자동 검증 도구인 Isabelle/ASVO는 Isabelle/Isar를 이용하여 구현되었으며, 프로토콜 명세와 ASVO 로직의 적용을 정확하게 할 수 있도록 개발되었다. 즉, 수동 검증을 통해서 야기될 수 있는 오류를 방지하고, 명확한 검증 결과를 얻을 수 있다는 장점이 있다. 하지만, Isabelle/Isar 언어로 번역되었으며, 구현 의존적인 변환을 필요로 하였다. Isabelle/ASVO에 대해서는 선행연구[16]에서 비교적 자세히 다루었으므로, 본 논문에서는 생략하기로 한다.

Ⅵ. Isabelle/ASVO를 이용한 검증 실험

6.1 실험 개요

Isabelle/ASVO 시스템을 사용하여 다음과 같은 프로토콜에 대해서 ASVO 로직 분석을 수행하였다.

① NSSK (Needham-Schroeder Shared Key) 프로토콜
② NSSK7 (Needham-Schroeder Shared Key 7) 프로토콜
③ NSPK (Needham-Schroeder Public Key) 프로토콜
④ NSPK2 (Needham-Schroeder Public Key 2) 프로토콜
⑤ NSPK3 (Needham-Schroeder Public Key 3) 프로토콜
⑥ DH (Diffie-Hellman) 프로토콜
⑦ STS (Station-to-Station) 프로토콜
⑧ STS2 (Station-to-Station 2) 프로토콜

먼저 대칭키 프로토콜로서 대표적인 NSSK 프로토콜을 분석하여, 잘 알려진 Denning-Sacco 공격[4]의 공격 포인트를 확인할 수 있었으며, GNY 로직의 recognizability와 같은 문제점[5]도 발견할 수 있었다. 따라서 이와 같은 두 가지 문제점을 해결한 NSSK7 프로토콜을 ASVO 로직의 추론 결과를 따라서 보완 설

계한 결과, ASVO 로직을 통해서 무사히 검증될 수 있었다. 또한 공개키 프로토콜로서 잘 알려진 NSPK 프로토콜을 검증하여, 인터리빙 공격에 대한 공격 포인트를 발견할 수 있었으며, 이를 보완한 프로토콜 NSPK2와 NSPK3을 설계하고 검증하였다. 한편 키 교환 프로토콜인 DH 프로토콜을 검증하고, 이에 대해서 인증 기능을 더한 STS 프로토콜과 STS2 프로토콜을 검증하고 함께 비교하였다. 본 논문에서는 자세한 검증 과정은 생략하며, 각 프로토콜의 RMA와 목표, 그리고 검증 결과에 대해서 설명하도록 한다.

6.2 대칭키 기반 프로토콜 분석 결과

대칭키 기반의 NSSK 프로토콜은 제 3의 신뢰기관(TTP) S를 통해서 프로토콜 참여자 A와 B는 서로 안전한 키분배와 인증을 이루어야한다.[9] 이것은 ASVO 목표에서 다음과 같이 인증을 위한 목표를 필요로 한다.

ASVO2': A believes B says $A \xleftrightarrow{K} B \wedge A$ believes $A \xleftrightarrow{K} B$

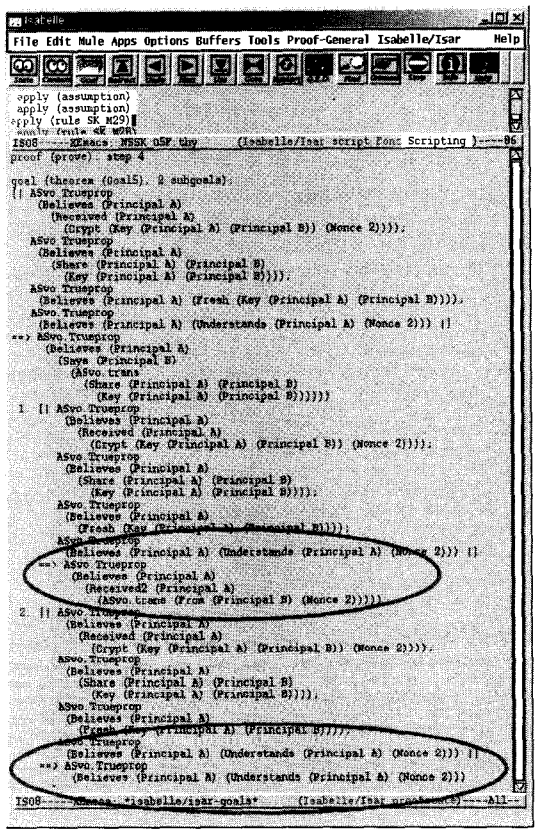
또한 더 나아가서 분배된 키의 신규성을 확인할 수 있어야한다.

ASVO5: A believes fresh(K)

이와 같은 목표를 각 참여자 A와 B에 대해서 설정한 후, ISA-RMA로부터 목표를 유도하는 각 theorem들을 정의하고 증명하면, [그림 7]과 같은 과정을 통해서 정확한 subgoal들을 얻을 수 있다. 각 subgoal들을 해결해 나가면 ASVO 로직을 통한 추론 목표에 도달한다. 하지만, [그림 7]의 2번 subgoal과 같이 불가능한 subgoal들이 주어지는데 요약하면 다음과 같다. 여기서 ---는 어느 공리 적용을 위해서 이와 같은 subgoal들이 필요했는지를 나타낸다.

A believes A understands N_B --- A5 공리
B believes Fresh(K_{AB}) --- A21 공리

하지만 이것은 추론 진행을 위해서는 필요하지만, 기본적인 ISA-RMA로부터는 얻을 수 없는 가정이다.



〔그림 7〕 Isabelle/ASVO 수행 결과

이것은 선행 연구에서 분석을 했듯이, Dumb Authentication 공격[16,5]과, Denning-Sacco 공격[4]을 허용하는 중요한 공격점이다. 따라서 우리는 이와 같은 subgoal을 제거하도록 재구성한 NSSK7 프로토콜을 다음과 같이 얻을 수 있었으며, 기본적인 ISA와 RMA만으로 ASVO 로직을 통한 증명이 가능했다.

$A \rightarrow B$: A
$B \rightarrow A$: B, N_{B1}
$A \rightarrow S$: A, B, N_A, N_{B_1}
$S \rightarrow A$: $\{N_A, B, K_{AB}, \{N_{B_1}, K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$
$A \rightarrow B$: $\{N_{B_1}, K_{AB}, A\}_{K_{BS}}$
$B \rightarrow A$: $\{A, B, N_{B_2}\}_{K_{AB}}$
$A \rightarrow B$: $\{N_{B_2} - 1\}_{K_{AB}}$

6.3 공개키 기반 프로토콜 분석 결과

NSPK 프로토콜은 Needham과 Schroeder에 이해서

제안되었으며, 공개키 기반의 프로토콜로서, NSSK 프로토콜과 함께 여러 분야에서 많은 관심의 대상이었다.[9] NSPK 프로토콜은 참여 주체가 서로 상대방의 인증된 공개키를 이용하여 논스 값을 교환하고, 논스 값 교환에 대한 신규성 확인이 될 경우, 서로의 온라인 상태를 확인하고 또한 공개키의 특성을 이용해서 서로를 인증하도록 하는 프로토콜이다. NSPK 프로토콜은 다음과 같이 구성된다.

$A \rightarrow B$: $\{N_A, A\}_{K_B}$
$B \rightarrow A$: $\{N_A, N_B\}_{K_A}$
$A \rightarrow B$: $\{N_B\}_{K_B}$

즉, 이와 같이 challege-response에 의존하는 프로토콜은 인증을 위해서 다음과 같은 ASVO 목표를 이루어야한다.

ASVO2: A believes B says $F(X) \wedge A$ believes A has X

하지만, 우리는 ASVO 로직 검증을 통해서 다음과 같은 subgoal을 갖게 되었다.

P121) A believes A received2 (N_A, N_B) from B --- A5 공리
P131) B believes B received2 N_B from A --- A5 공리

역시 ISA-RMA로부터 추론할 수 없는 subgoal이며, 이것을 통해서 주체 A와 B 모두 메시지의 근원지에 대해서 논리적으로 확인할 수 없음을 알 수 있다. 이로 인하여 가능한 공격은 잘 알려진 Lowe의 공격이다.[7]

따라서 ASVO 로직을 통한 분석을 위해서, 다음과 같이 NSPK2 프로토콜과, 해당 프로토콜을 위한 외부 공리를 다음과 같이 정의하였다.

$A \rightarrow B$: $\{A, N_A\}_{K_B}$
$B \rightarrow A$: $\{B, N_A, N_B\}_{K_A}$
$A \rightarrow B$: $\{A, N_B\}_{K_B}$

A23a2) ($\mathrm{PK}_{\phi}(\mathrm{P}, \mathrm{K}) \wedge P$ received $\{Q, X\}_K$) $\Rightarrow P$ received2 X from Q

즉, NSPK 프로토콜과 달리 NSPK2 프로토콜에서는 모든 메시지에 송신자의 신원을 포함하도록 하였으며, 이에 대한 문맥적 분석 공리 A23a2를 정의하였다. 하지만 이와 같은 분석은 문맥의 특성에 의존적이므로, ASVO 로직에는 기본적으로 존재하지 않는 공리이다. 물론 이와 같은 외부 공리마저도 필요치 않은 프로토콜은 NSPK3이다.

$A \rightarrow B$: $\{N_A, B\}_{K_A^{-1}}$
$B \rightarrow A$: $\{N_A - 1, N_B\}_{K_B^{-1}}$
$A \rightarrow B$: $\{N_B - 1\}_{K_A^{-1}}$

즉, NSPK 프로토콜과 같은 메시지 구조를 갖지만, 사용된 키가 상대방의 공개키가 아닌 자신의 개인키이다. 다시 말하면 전자서명을 사용한 것이다. 이와 같은 프로토콜은 인증에 관한 목표는 이룰 수 있지만, 키 분배에 대해서는 목표를 이룰 수 없다. 이에 대해서도 역시 ASVO 로직으로 명확히 분석되었다.

6.4 키교환 프로토콜 분석 결과

Diffie-Hellman 키교환 프로토콜은 기본적으로 키 교환만을 위해서 설계되었다. DH 프로토콜을 ASVO 로직의 구문에 맞게 기술하면 다음과 같다.

$A \rightarrow B$: K_a
$B \rightarrow A$: K_b

여기서 K_a나 K_b와 같이 키의 아래첨자가 소문자인 경우는, Diffie-Hellman 공개키임을 의미하며[13], 결과적으로 K_{ab}는 Diffie-Hellman 합의 키를 의미한다[14]. DH 프로토콜은 기본적으로 참여자 서로에 대한 인증뿐만 아니라 키에 대한 인증 또는 확인 과정을 포함하지 않으므로, 다음과 같이 키 공유에 대한 목표를 설정할 수 있다. 즉, ASVO3 목표에 해당하는 비검증된 키 분배를 의미한다.

ASVO3 : A believes A has $K \wedge A$ believes fresh (K)

13) 예를 들면, $K_a = g^x \bmod p$이고 $K_b = g^y \bmod p$이다.
14) 예를 들면, $K_{ab} = g^{xy} \bmod p$이다.

것은 참여자가 서로 문맥상 공유 키라고 할 수 있는 값 K_{ab}에 대해서 신규성과 소유 사실을 확인하고 신뢰할 수 있지만, 서로 상대방과 통신하기에 안전한 키에 해당하는지는 확인할 수 없는 상태이다. 즉, 따라서 man-in-the-middle 공격과 같은 공격에 노출될 수 있다. STS(Station-to-Station) 프로토콜은 DH 프로토콜에 전자서명을 통한 인증 기능을 첨가한 대표적인 프로토콜이다.[8]

$A \rightarrow B$: K_a
$B \rightarrow A$: $K_b,\ \{\{K_b, K_a\}_{K_B^{-1}}\}_{K_{ab}}$
$A \rightarrow B$: $\{\{K_a, K_b\}_{K_A^{-1}}\}_{K_{ab}}$

즉, STS 프로토콜은 기본적으로 Diffie-Hellman 키 교환을 통한 키 합의 과정을 포함하므로 다음과 같이 검증된 키 분배 목표를 설정할 수 있다.

ASVO4: A believes $A \xleftrightarrow{K} B \wedge A$ believes fresh (K)

하지만 프로토콜 상호 확인과, 명확한 인증을 위해서 다음과 같은 목표를 설정할 경우, ASVO 로직을 통해서 검증할 수 없다.

ASVO6: A believes B says $A \xleftrightarrow{K} B$

이것은 STS 프로토콜이 수행되어도, 각 주체가 서로 상대방이 자신을 신뢰하고 있는지 확인할 수 없다는 것이다. 따라서 다음과 같은 Lowe의 공격점을 의미한다고 할 수 있다.[8]

$A \rightarrow E(B)$: K_a
$E \rightarrow B$: K_a
$B \rightarrow E$: $K_b,\ \{\{K_b, K_a\}_{K_B^{-1}}\}_{K_{ab}}$
$E(B) \rightarrow A$: $K_b,\ \{\{K_b, K_a\}_{K_B^{-1}}\}_{K_{ab}}$
$A \rightarrow E(B)$: $\{\{K_a, K_b\}_{K_A^{-1}}\}_{K_{ab}}$

여기서 우리가 ASVO 로직을 통해서 살펴본 바와 같이, 주체 A는 ASVO6를 만족할 수 없으므로, 주체 B를 가장한 공격자 E의 메시지로부터 B와의 키 공유에 대한 신뢰만 얻어낼 뿐이다. 구체적으로 B 역시 같은 키를 신뢰하고 있는지 여부에 대해서는 확인할 수 없다. 또한, 주체 B의 경우, 마지막 메시지를 전송받지 않았으므로, 상대방 E와의 어떠한 신뢰도 얻을 수 없다. 이와 같이 ASVO 로직을 통한 분석은 기존의 SVO 로직에서 명확히 지적하지 않았던 Lowe의 공격을, ASVO 로직의 goal과 공리에 근거하여 명확하게 밝혀낼 수 있다.

ASVO 로직의 논리적인 분석을 통과하기 위해서는, 주어진 subgoal들을 바탕으로 STS 프로토콜을 수정해야 하며, 다음과 같은 STS2 프로토콜을 얻을 수 있다.

$A \rightarrow B$: $A, \{K_a\}_{K_A^{-1}}$
$B \rightarrow A$: $B, \{K_b\}_{K_B^{-1}},\ \{A, B, K_a\}_{K_{ab}}$
$A \rightarrow B$: $\{B, A, K_b\}_{K_{ab}}$

즉, 주체 B가 첫 메시지에서 K_a에 대한 신뢰를 얻도록 하며, 주체 A 역시 이와 같은 신뢰를 바탕으로 ASVO6 목표를 두 번째 단계에서 달성할 수 있도록 한다. 이와 같은 프로토콜은 Lowe의 공격에 대해서 안전하다. 결과적으로 우리는 STS 프로토콜에 대한 ASVO 로직 분석을 통해서 메시지에 대한 이해와 신뢰의 논리적인 순서도 프로토콜의 안전성에 있어서 매우 중요하다는 것을 알 수 있었다.

Ⅶ. 결 론

본 논문에서는 SVO 로직에 바탕을 둔 ASVO 로직을 새로이 제안하고, 대표적 유형의 프로토콜에 대한 검증 결과를 요약하였다. ASVO 로직은 SVO 로직과 구문적/의미적인 면에서 유사하지만, SVO 로직의 CA, IA 과정을 제거하는 등 많은 차이가 있다. 또한 자동 검증을 지원하는 도구 Isabelle/Isar를 구현하였다. 따라서 검증자는 프로토콜에 대한 복잡한 이해없이도, 정확한 추론을 통한 프로토콜 검증을 할 수 있다. 다만, 검증 가능한 프로토콜의 종류는 심볼릭 언어로 표현 가능한 인증 및 키 분배 프로토콜이다. 현재 구현된 Isabelle/ASVO 시스템은 Isabelle/Isar의 비교적 편리하고 쉬운 해석 언어 환경에서 ASVO 로직을 이용한 프로토콜 검증을 지원한다. 따라서 상대적으로 효율적인 tactic의 마련이 어려웠다. 향후 연구에서는 Isabelle/Pure와 같은 환경에서 ML을 직접 이용하여 재개발하여 효율적인 tactic의 마련과 함께 자동화 기능을 향상시킬 필요가 있다. 또한 프로토콜

을 사용자가 쉽게 입력하기 위한 번역 언어의 마련도 필요하겠다.

참 고 문 헌

[1] M Abadi and M. Tuttle, "A semantics for a logic of authentication", In Proc. of the Tenth Annual ACM Symposium on Principles of Distributed Computing, pp.201~216, 1991.

[2] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication", Technical Report SRC RR 39, Digital Equipment Corporation, Systems Research Center, February 1990. Also published in ACM Transactions on Computer Systems, vol.8, no.1, pp.18~36, February 1990.

[3] A. Dekker, "C3PO: a tool for automatic sound cryptographic protocol analysis", In Proc. of the IEEE Computer Security Foundation Workshop, June 2000.

[4] D. Denning and G. Sacco, "Timestamps in key distribution protocols", Communications of the ACM, vol.24, no.8, pp.533~536, August 1981.

[5] L. Gong, R. Needham, and R. Yahalom, "Reasoning about belief in cryptographic protocols", In Proc. of the IEEE Symposium on Research in Security and Privacy, pp.234~248, 1990.

[6] S. Gritzalis, D. Spinellis, and P. Georgiadis, "Security protocols over open networks and distributed systems: formal methods for their analysis, design, and verification", Computer Communications, vol.22, no.8, pp.695-707, May 1999.

[7] G. Lowe, "Breaking and fixing the Needham-Schroeder public-key protocol using FDR", Software-Concepts and Tools, vol.17, pp.93~102, 1996.

[8] G. Lowe, "Some new attacks upon security protocols", IEEE Computer Security Foundations Workshop, pp.162~169, 1996.

[9] R. Needham and M. Schroeder, "Using encryption for authentication in large networks of computers", Communications of the ACM, vol.21, no.12, pp. 993~999, 1978.

[10] T. Nipkow, L. Paulson, and M. Wenzel, "Isabelle/HOL", Lecture Notes in Computer Science, Vol. 2283, Springer-Verlag, 2002.

[11] P. Syverson and P. van Oorschot, "On unifying some cryptographic protocol logics", In Proc. of the IEEE Symposium on Research in Security and Privacy, pp.14~28, 1994.

[12] P. Syverson and P. van Oorschot, "A unified cryptographic protocol logic", NRL Publication 5540~227, Naval Research Lab, 1996.

[13] P. Syverson and Iliano Cervesato, "The logic of authentication protocols", Lecture Notes in Computer Science, Vol.2171, Springer-Verlag, 2002.

[14] P. van Oorschot, "Extending cryptographic logics of belief to key agreement protocols", In Proc. of the ACM Conference on Computer Communications Security, pp.232~243, 1993.

[15] R. Goldblatt, "Mathematical Modal Logic: a View of its Evolution", http://www.mcs.vuw.ac.nz/~rob, February, 2002.

[16] 권태경, 양숙현, 김승주, 임선간, "암호프로토콜 논리성 자동 검증에 관한 연구", 한국정보보호학회 논문지 vol.13, no.1, pp.115~130, February, 2003.

[17] Taekyoung Kwon, Seongan Lim, "Automation-considered Logic of Authentication and key distribution", WISA, Lecture Notes in Computer Science, 게재예정, 2003.

〈著者紹介〉

권 태 경 (Taekyoung Kwon) 종신회원
1992년 : 연세대학교 컴퓨터과학과 졸업
1995년 : 연세대학교 컴퓨터과학과 석사
1999년 : 연세대학교 컴퓨터과학과 박사
1999년~2000년 : U.C. Berkeley Post-Doc.
2001년~현재 : 세종대학교 컴퓨터공학부 소프트웨어공학과 조교수, 정보보호학회 편집위원, TTA 암호분과 특별위원
<관심분야> 정보보호, 암호프로토콜, 네트워크 보안 등

임 선 간 (Seongan Lim) 종신회원
1985년 : 동국대학교 수학과 학사
1987년 : 서울대학교 수학과 석사
1995년 : Purdue 대학교 수학과 박사
1999년~현재 : 한국정보보호진흥원 암호기술팀 팀장
<관심분야> 암호프로토콜, 정보보호 등

박 해 룡 (Haeryong Park)
1999년 : 전남대학교 수학과 이학사
2001년 : 서울대학교 대학원 수학과 이학석사
2000년~현재 : 한국정보보호 진흥원(KISA) 연구원
<관심분야> 키관리, 암호프로토콜, 전자화폐 등