情 報 保 護 學 會 誌
第17卷 第6號, 2007. 12

# 고등급 평가 기술 : 정형화 기법 개요

심재환*, 김현석*, 유희준**, 최진영*

요 약

정보통신과 인터넷 기술의 발전으로 실생활에서 정보의 이용이 용이해 지면서, 쉽게 정보를 얻기 위한 정보에 대한 공유와 검색에 관련한 기술이 발전하게 되었다. 그 결과로 정보는 쉽게 얻을 수 있게 되었지만, 개인정보 유출, 악의적인 시스템 파괴 등 역기능 빠른 속도로 늘어나고 있다. 이에 따라 국제적으로 정보보호의 중요성대 대두 되고 보안에 대한 필요성이 극대화 되었으며, 시스템 보안성의 신뢰도를 일관성 있게 평가하기위하여 공통평가기준이 제정되었다. 특히, 높은 신뢰도를 요구하는 보안기능에 대해서는 수학적, 논리적 기반의 명확한 명세와 엄밀한 증명을 수행하는 정형기법을 사용한 경우에만 고등급을 획득할 수 있다. 본고에서는 공통평가기준과 정형기법의 상관관계를 설명하고, 고등급 평가에 관련된 국외 동향을 기술한 후, 간단한 예를 들어 보안정책모델의 정형화 방법에 대하여 기술하였다.

## Ⅰ. 서 론

20세기 후반부터 정보통신 기술과 인터넷 기반 인프라가 급속도록 발전하면서 인터넷은 인간의 삶의 질에 큰 영향을 미치고 있다. 인터넷을 통해 상상할 수 없을 정도의 많은 정보가 유통되고 있으며, 삶을 보다 편리하고 윤택하게 만들어 주고 있다. 이를 위해서 사용자들의 편리하게 정보를 획득하게 하기 위해서 정보 검색 및 정보 공유에 관련된 기술이 급속하게 발전하고 있다. 이러한 기술의 발달은 정보를 쉽게 획득하고 가공할 수 있도록 해주었지만, 정보의 오·남용과 개인정보 유출, 저작권의 침해, 악의적인 정보시스템 파괴 등 인터넷의 역기능은 무시할 수 있는 범위를 이미 넘어서고 있는 실정이다. 이에 따라 정보보호의 중요성은 지속적으로 강조되고 있다. 이러한 상황에서 사용자는 보안 관련 프로그램들의 보안기능이 신뢰할 수 있는 수준의 명확한 보안성을 가지고 있는지 여부에 관심을 가지게 되었으며, 국제적으로 일관성 있는 방법론으로 보안 기능의 신뢰도를 평가하는 것은 매우 중요한 이슈로 등장하게 되었으며, 북미지역의 TCSEC, 유럽의 ITSEC과 국내의 K등급과 같이 보안제품을 평가하기 위한 기준 및 제도를 마련하게 되었다. 하지만, 보안상의 위협이 국가내의 문제에서, 국가와 국가 간의 문제로 커져가면서, 국제적으로 동일한 기준으로 보안제품 보안성에 대한 신뢰도를 평가하는 것이 중요한 이슈가 되었다. 이를 위해 국제 표준인 공통평가기준(Common Criteria : CC)이 만들어졌다. 이 중 5단계 이상의 고 보증등급은 높은 수준의 보안 시스템을 개발하는 업체들에게 필수적인 보증등급이며, 이러한 고 보증등급을 획득하기 위해서 제품의 개발클래스 부분을 정형기법을 사용하여 명세하도록 공통평가기준에서는 요구하고 있다[1].

본 고에서는 보안시스템의 고 보증등급획득을 위해 공통평가기준에서 요구하고 있는 정형기법과 그 적용예를 통해 고 등급평가에서의 정형화 기법에 대해 논할 것이다. 본고의 구성은 아래와 같다.

2장에서는 공통평가기준과 공통평가기준에서 보증등급에 따른 정형화 정도 및 정형화 될 문서에 대해서 살펴볼 것이다. 3장에서는 현재까지 국외에서 진행된 고등급 평가에 관련된 작업을 설명하겠다. 4장에서는 정형 명세언어에 대한 간략한 소개를 할 것이다. 5장에서는 보안정책모델에 대해서 정형기법을 적용 시에 반영되어야 하는 문서의 구성 및 내용에 대해 살펴보고 6장에서는 정형 명세언어인 Z를 이용하여 보안정책모델을 개발하기 위한 간략한 사례를 보여준 후 결론을 맺도록 하겠다.

* 고려대학교 정보통신대학 컴퓨터 · 통신공학부 (jhsim@formal.korea.ac.kr,hskim@formal.korea.ac.kr,choi@formal.korea.ac.kr)
** 한국정보보호진흥원 보안성평가단 (hjyoo@kisa.or.kr)

## Ⅱ. 공통평가기준과 정형기법

공통평가기준(Common Criteria for Information Technology Security Evaluation)은 보안관련 컴퓨팅 기술의 평가를 위한 국제 표준이다. ITSEC[6], TCSEC 등 유사 종류의 표준을 범국가적으로 통합하고자 하는 목적으로 제정되었다. 공통평가기준은 보증 수준에 따라 7단계의 평가보증등급(Evaluation Assurance Level)을 정의하고 있다. 특히 EAL5에서 EAL7까지는 고 보증등급으로 불린다.[1][2][4]

CC 기반의 평가 • 인증 제도를 사용하고 있는 국가들은 서로의 평가 • 인증 결과를 상호 인정해주기 위해서 공통평가기준 상호인정협약(Common Criteria Recognition Arrangement : CCRA)를 맺고 있다. CCRA에는 인증서를 발행하는 국가들(Certificate Authorizing Paticipants : CAP)과 인증서를 수용하고 있는 국가들(Certificate Consuming Participants)로 구성되어 있다. 우리나라는 2006년 CAP로 가입하여 활동하고 있으며, 현재, CCRA에는 CAP 12개국과 CCP 12개국으로 구성되어 활발한 활동을 하고 있다.

공통평가기준에서는 제품을 평가하기 위해 사용자 혹은 개발자가 제시하는 보안 요구사항, 즉 반드시 만족해야 되는 요구사항의 표준을 보호 프로파일(Protection Profile : PP)이라는 개념으로 소개한다. 평가대상(Target of Evaluation, TOE)은 평가의 대상이 되는 전체 혹은 부분 시스템을 의미하는 개념이다. 보안목표(Security Target, ST)는 평가자가 평가의 기반으로 사용할 정보로서 평가대상에 대해 보안목표가 만족됨을 보이는 인증과 이에 관련된 문서를 결과로 산출한다. 공통평가기준의 평가보증등급은 [표 1]과 같다.

[표 1]에서 보는 바와 같이 고 보증등급인 EAL5에서 EAL7의 경우에 준 정형기법(Semiformal) 혹은 정형기법(Formal)을 이용하여 평가대상을 설계하고 검증하고, 테스트하기를 요구하고 있다. 따라서 보안시스템의 개발주체는 고등급의 보증등급을 획득하기 위해서는 정형기법의 적용이 필수적이다.

또한, 공통평가기준은 인증을 위한 요구사항을 정의하고 있는데, 이 요구사항들은 개발(Development), 형상관리(Configuration Management), 시험(Testing) 등의 클래스로 분류되어있다. 이 중 실제 보안 시스템의 개발 단계에서 작성되어지고 정형기법과 관련된 문서는 개발 클래스이다.

〔표 1〕 공통평가기준의 평가보증등급

| | |
|---|---|
| EAL7 | Formally verified design and tested |
| EAL6 | Semiformally verified design and tested |
| EAL5 | Semiformally designed and tested |
| EAL4 | Methodically designed, tested and checked |
| EAL3 | Methodically tested and checked |
| EAL2 | Structurally tested |
| EAL1 | Functionally tested |

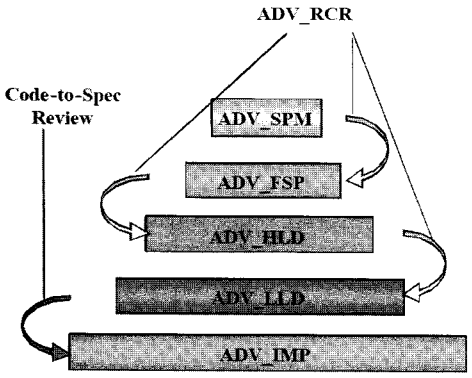
이러한 개발클래스는 TOE의 보안기능 전반에 걸친 설계, 구현 및 검증에 해당하는 부분을 작성한 문서로 시스템의 기능상의 신뢰도를 평가하기 위해서 중요한 부분을 차지하게 된다. 개발클래스를 살펴보면 보안기능의 기본 설계, 보안정책모델과 기능명세, 이들 간의 일치성은 정형적인 방법으로 기술해야지만 최상위 등급인 EAL7을 획득할 수 있다. 이는 TOE의 보안 기능에 대해서 설계단계부터 기능구현까지 일련의 개발 과정이 수학적, 논리적으로 일관성 있게 작업되었음을 의미하게 된다. [표 2]는 고 보증등급에서의 개발 클래스의 요구사항을 나타내고 있다.

[표 2]에서 보는 바와 같이 EAL5 이상의 고 보증 등급을 위한 문서는 준 정형기법 혹은 정형기법의 사용을

〔표 2〕 개발 클래스에서의 요구사항

| Assurance Class | Assurance Family | Evaluation assurance level | | |
|---|---|---|---|---|
| | | EAL5 | EAL6 | EAL7 |
| Develop-ment | ADV_FSP (기능 명세) | SF | SF | F |
| | ADV_HLD (기본 설계) | SF | SF | F |
| | ADV_IMP (구현의 표현) | I | I | I |
| | ADV_INT (내부 설계) | I | I | I |
| | ADV_LLD (상세 설계) | I | SF | SF |
| | ADV_RCR (일치성 입증) | SF | SF | F |
| | ADV_SPM (보안정책모델) | F | F | F |

※ (F : Formal, SF : Semiformal, I : Informal)




〔그림 1〕 Design Assurance Architecture

요구하고 있다. 특히 보안정책모델은 [그림 1]에서 보는 바와 같이 모든 개발 클래스 문서의 기본이 되는 문서로서, EAL5 이상의 모든 고 보증등급에서 정형기법의 사용이 요구되어지고 있다.[3]

하지만, 공통평가기준에서 요구하고 있는 정형기법은 특정한 기법이나 언어를 강제하고 있지 않다. 개발하고자 하는 보안시스템에 적합한 기법 및 언어를 선정하여 보안시스템의 개발에 이용할 수 있다. 4 장에서는 적절한 기법의 선정을 위해 다양한 정형 명세언어를 살펴보도록 하겠다.

## Ⅲ. 국외 고등급 평가 사례

국제적으로 정보보호의 중요성이 높아가면서 신뢰도가 높은 보안제품의 필요성 역시 함께 증가하게 되었다. 이러한 이유로 CC 기반의 평가 • 인증 제도를 사용하고 있는 국가들에서는 고등급 평가에 관련된 연구가 2001년도부터 지속적으로 진행되고 있으며, 현재까지 국제적으로 EAL5 이상의 고등급을 획득한 제품은 40여개 정도가 된다.


〔그림 2〕 제품별 고등급 평가 분포도

고등급 평가를 받은 제품들을 살펴보면 전체 60%정도의 제품들이 ICs, Smart Card 혹은 Smart Card와 연관된 디바이스와 시스템들이 차지하고 있으며, 나머지 부분을 운영체제 제품들과 네트워크 장비들이 차지하고 있다. 국가별로 보면, 독일이 60%, 미국이 15% 정도를 차지하고 있다.

국가 혹은 제품별 정보보호정책을 살펴보면 위의 결과와 밀접한 결과가 있음을 알 수 있다. 유럽에서 활발한 연구와 개발이 이루어지고 있는 스마트카드의 경우는 대부분의 제품이 다목적 IC 카드용 운영체제인 MULTOS를 사용하고 있으며, MULTOS 개발자 연합인 MAOSCO는 내부정책으로 MULTOS를 사용하는 COS는 ITSEC E6이상의 인증을 받을 것을 요구하면서, 인증 수준에 따라서 카드키 발급수량을 제한함으로써 개발/공급자가 반드시 인증을 받아야만 한다. MAOSCO는 인증기준을 ITSEC에서 CC로 변환하면서, EAL5 이상의 경우는 스마트카드의 무제한 발급이 가능하나 EAL4일 경우 100만장으로 생산을 제한하고 있는 실정이다.

미국의 국가 조달 정책을 보면, [표 3]과 같이 정보의 가치 및 위협에 따라서 3단계로 SML(Strength of Mechanism Levels)과 EAL(Evaluation Assurance Levels)을 부여하여 정보보호제품을 사용할 것을 명시하고 있다.

이와 같이, 국가나 제품별로 강력한 보안기능이 요구되는 경우 보다 강력한 신뢰도와 보증 요구사항을 요구하고 발생하고 있는 실정이다.

〔표 3〕 미국의 보안제품 조달 정책 일부

| Robustness | 주요내용 | EAL |
|---|---|---|
| Basic | 우수한 상용제품과 동등한 보안 서비스 및 메커니즘 | EAL2 이상 |
| Medium | Basic보다는 높은, 추가적인 보호계층을 제공하는 보안서비스 및 메커니즘 | EAL4 이상 |
| High | 가능한 가장 엄격한 보호와 면밀한 보안 대책을 제공하는 보안서비스와 메커니즘 | EAL6 이상 |

〔표 4〕 정형명세 언어와 도구의 예

| 기법 | 연구(개발)기관 | 명세/검증 | 적용 예(분야) |
|---|---|---|---|
| Z | ESPRIT & U.K government | 명세/검증 | Tektronix oscilloscopes |
| VDM | IFAD | 명세 | Railway interlocking Systems |
| MSC | ITU-T | 명세 | Switching System |
| SCR | NRL | 명세 | Reactor Protection System |
| SPIN | Bell Lab. | 명세/검증 | OSI 7-layer |
| ACSR | UPENN | 명세/검증 | Real-time System |
| CSP | C.A.R Hoare | 명세/검증 | TMN protocol |
| B | J.-R. Abrial | 명세 | SACEM |

## Ⅳ. 정형 명세언어 및 도구

### 4.1. 정형 명세 언어와 그 특징

[표 4]에서 보는 것과 같이 정형기법에 사용하는 언어 및 도구는 매우 다양하다. Z는 일차 논리와 집합론에 기반을 둔 간결하고 수학적인 명세 언어이다. MSC는 Message Sequence Chart의 약자로써 ITU-T에서 통신 프로토콜 설계의 표준으로 채택하여 사용하고 있다. 이는 SDL(Specification and Description Language)의 대표적인 도구로서 각종 통신 프로토콜의 설계에 매우 활발히 사용 중이다. SPIN은 Bell Lab.에서 개발한 소프트웨어 검증 도구로서 통신 프로토콜의 검증에 매우 유용하게 사용되고 있다. 이 도구는 Promela라는 명세 언어를 입력으로 사용하며 선형시제논리(LTL)를 이용하여 정형검증을 실행한다. ACSR은 CCS(Calculus for Communicating Systems)에 기반한 프로세스 대수(Process Algebra)로서 시간, 자원, 우선순위 동시성 등 실시간 시스템에 필요한 여러 개념을 포함한다. 실시간 시스템의 정확성은 계산 결과가 얻어지는 시간과 밀접한 관련이 있으므로 어떤 사건이 발생하는 시간을 파악할 수 있는 방법을 프로세스 대수에서 제공하는 것이 필요하다. 이 외에도 현재 수백 여 가지의 정형기법 언어와 도구가 개발되어 각각의 목적에 맞게 사용되어 지고 있다. 특히 정형 명세 언어 중 Z는 수학적인 표현을 통해 간결하고 애매모호함이 없는 정확한 명세가 가능하다는 특징을 가지고 있다. 또한 스마트카트 운영체제인 Gemplus Multos를 포함하여 많은 공통평가기준인증을 위한 명세과정에 이용되어지고 있다. 다음에서 정형 명세 언어 Z에 대해서 살펴보도록 하겠다.

### 4.2. 정형 명세 언어 Z

Z 언어는 일차논리(First-Order Logic)와 집합론(Set Theory)과 같은 수학적인 기반을 가지고 있고, 이로 인해서 명세에 많은 이득을 가지고 있다. 예를 들면, 이러한 수학적인 표현은 간결하고, 애매모호함이 없기 때문에 정확한 명세를 할 수 있다. 따라서 이러한 명세를 보고 이해하기가 쉽다.

Z는 1970년대 후반에서 1980년대 초반에 걸쳐서 영국의 옥스퍼드 대학(Oxford University)의 프로그래밍 그룹(Programming Research Group)의 Jean-Raymond Abrial, Bernard Sufrin과 IbSØrensen에 의해서 개발되었다. Z 언어는 개발 초기서부터 학술적인 범위를 벗어나서 실 시스템 명세에 사용되었다. 특히, IBM Hursley는 Z를 이용해서 이미 그들이 성공을 거둔 시스템인 고객 제어 정보 시스템(CICS : Customer Information Control System)을 재 명세(re-specification)하였다. 이 예는 Z의 발전에 매우 유용한 효과를 주었다. 이 결과 Z 언어는 산업 환경에서 커다란 소프트웨어 시스템을 명세 하는 실질적인 결과를 내면서 성장하였다. 이러한 과정을 거치면서 많은 결과를 쌓게 되었고, 결국 1989년에 Spivey에 의해서 이론적인 고안으로 Z 표준 언어가 정의되었다.

## Ⅴ. 정형기법을 이용한 보안정책모델

보안정책모델에 대한 문서는 모든 개발클래스 문서의 기초가 되며, 고 보증등급 보안정책모델은 정형기법이 요구됨은 앞서 서술한 바와 같다. 본 장에서는 정형기법을 이용한 보안정책문서의 문서의 구조와 포함될 내용에 대해서 설명할 것이다.

보안정책문서는 다음과 같이 구성 되어 질 수 있다.

1) 보안정책의 정형모델 (formal model of security policy)

이 항목은 보안정책에 대해서 정형화된 모델을 명세

하는 것이다. 이것은 데이터모델, 상태모델, 정책모델로 나누어진다. 이 세 가지 모델은 다음과 같다.

- 데이터모델 : 데이터모델은 보안정책의 기본적인 기능들과 이를 표현하기 위한 데이터 타입들을 명세한다.
- 상태모델 : 상태모델은 평가대상이 만족하고자 하는 기본적인 안전한 상태를 명세한다. 이것은 데이터모델에서 명세 된 기능들과 이 기능들 사이의 불변식(invariant)을 포함한다.
- 정책모델 : 정책모델은 보안목표명세서(Security Target)에 명시된 요구사항을 만족시키기 위한 보안 기능들의 전제조건(Precondition)을 명세한다.

2) 보안기능의 정형명세 (formal specification of security function)

이 항목은 평가대상에게 요구되는 보안기능(Security Function)에 대해서 정형명세 하는 부분이다.

3) 보안정책에 대한 증명 (proof of formal model of security policy)

이 항목은 보안정책모델의 일치성(Consistency)과 완전성(Completeness)을 보증하기 위해 작성되어진다. 이 때, 일치성과 완전성은 다음과 같다.

- 일치성 : 일치성은 주어진 명세가 논리적으로 건전(Sound)하고, 모순이 없음을 의미한다. 일치성을 증명하기 위해서는, 첫째, 데이터 타입들간의 일치성을 보여야하고, 둘째, 기능들 사이의 일치성을 보여야하고, 마지막으로 주어진 모델의 상태모델과 초기화 모델과의 일관성을 보여주면 된다.

- 완전성 : 완전성은 주어진 모델이 자신이 설정한 도메인 내에서만 동작한다는 것을 의미한다. 즉, 정책모델은 정의된 모든 입력들에 대해서 완전히 명세 되어 있어야만 한다. 정책모델의 완전성을 보장하기 위해서는 모델의 수행(Operation)이 설정된 도메인 외에는 동작하지 않음을 전제조건을 이용하여 보여야만 한다. 만약 모델의 수행이 설정된 도메인 외에서 동작한다면, 모델은 정의되지 않은 상태, 즉, 불안전한 상태가 될 것이다.

## Ⅵ. 정형기법 Z를 이용한 보안정책문서 사례

이 장에서는 정형기법 Z를 이용한 보안정책모델의 사례를 보여줄 것이다. 사례를 위해 [5]에서 소개된 운영체제의 스케줄러를 평가대상(TOE)로 선정하고, 한 가지 보안특성을 추가 하였다. 이 운영체제가 탑재된 시스템에 대한 기본적인 설명은 다음과 같다. 첫째, 한 개의 CPU만을 사용하여 동시에 하나의 프로세스만이 CPU 자원을 사용할 수 있다고 가정하였다. 둘째, 스케줄러는 어떤 프로세스를 수행시킬지, 또 언제 수행시킬 지를 결정한다. 셋째, 프로세스는 스케줄 되기 전에 반드시 생성되어 있어야 하고, 프로세스는 자신의 일을 다 처리한 뒤 소멸되어야 한다. 넷째, 프로세스는 실행(current), 준비(ready), 대기(blocked)의 세 가지 상태를 갖는다. 다섯째, 평가대상인 운영체제의 스케줄러의 보안 요구사항으로 "생성 권한이 부여된 프로그램만이 생성될 수 있다."고 가정하였다. 마지막으로, 이 시스템은 생성권한이 부여된 프로그램과 생성 권한이 부여되지 않은 프로그램을 구별할 수 있는 메커니즘이 존재한다고 가정하였다.

[4]에는 스케줄링에 관한 여러 가지 기능 명세가 존재하지만, 여기서는 앞에서 설정한 보안 요구사항과 관계있는 프로세스 생성 기능에 대해서만 고려할 것이다.

이제부터 Ⅳ장에서 언급된 과정과 Z를 이용하여 보안정책모델을 작성하고 증명하는 과정을 살펴보자. 이 과정을 통해 보안기능이 보안 요구사항을 만족함을 보일 수 있다.

### 6.1. 보안정책의 정형모델

- 데이터모델

본 시스템은 n개의 프로세스를 처리할 수 있다. 여기서 n은 자연수 이다.

$$| \; n : \mathbb{N}$$

각각의 프로세스는 식별자 pid를 가지며, pid는 1과 n사이의 값이다.

$$PId == 1..n$$

생성된 프로세스가 없을 경우(null process) 자연수 0을 이용해서 식별자(null Pid)를 표현한다.

$$nullPId == 0$$

모든 경우의 식별자를 표현하기 위해서 'optional pid'를 정의한다. 'optional pid'는 실제 프로세스의 식별자와 'null pid' 모두 될 수 있다.

$$OptPId == PId \cup \{nullPId\}$$

PROGRAM은 보조 저장장치에 저장된 프로그램을 표현하며, 기본 타입으로 표시된다.

$$[PROGRAM]$$

다음의 집합은 생성 권한이 부여된 프로그램과 권한이 부여되지 않은 프로그램을 구별할 수 있는 메커니즘을 표현하였다. 집합 'Authorized'는 생성 될 권한을 부여받은 모든 프로그램의 집합이고, 운영체제의 생성기능은 집합 'Authorized'를 통해 프로그램의 권한을 구별한다.

$$| \; Authorized : \mathbb{P}\, PROGRAM$$

- 상태모델

스케줄러의 상태는 모든 프로세스를 네 가지의 상태로 분류한다. 프로세스는 실행(current), 준비(ready), 대기(blocked), 미생성(free) 상태 중 하나가 된다. 미생성(free) 상태는 현재 생성되지 않은 프로세스의 집합을 나타낸다. 즉, 운영체제에 의해 생성되지 않은 프로세스의 집합이다.

*Scheduler*

$$current : OptPId$$
$$ready : \mathbb{P}\, PId$$
$$blocked : \mathbb{P}\, PId$$
$$free : \mathbb{P}\, PId$$

---

$$\langle \{current\} \backslash \{nullPId\},$$
$$ready,$$
$$clocked,$$
$$free \rangle \text{ partition } PId$$

- 정책모델

앞서 언급된 바와 같이 생성 권한을 부여받은 프로그램만이 시스템에서 생성될 수 있다. 따라서 프로그램으로부터 프로세스를 생성하기위한 전제조건은 다음과 같은 스키마(Schema)로 표현될 수 있다.

*PreCreate*

$$Scheduler$$
$$g? : PROGRAM$$

---

$$free \neq \emptyset$$
$$g? \in Authorized$$

## 6.2. 보안기능의 정형명세

스케줄러가 초기 상태일 때, 모든 프로세스는 미생성(free) 상태이고, 실행상태인 프로세스는 존재하지 않는다.

*SchedulerInit*

$$Scheduler'$$

---

$$current' = nullPId$$
$$ready' = \emptyset$$
$$blocked' = \emptyset$$
$$free' = PId$$

프로세스가 생성되었을 때, 프로세스는 미생성 상태에서 준비상태가 되며, 미생성 상태의 프로그램이 존재해야 운영체제는 생성 기능을 수행할 수 있다.

*Create*

$$PreCreate$$
$$Scheduler'$$
$$p! : PId$$

---

$$current' = current$$
$$ready' = ready \cup \{p!\}$$
$$blocked' = blocked$$
$$free' = free \backslash \{p!\}$$
$$p! \in free$$

## 6.3. 보안정책에 대한 증명

보안정책모델을 정형명세하고 이를 검증하기 위해서는 명세의 정확성(Correctness), 명세의 일관성 (Consistency), 명세의 완전성(Completeness)를 보여야 한다. 정확성은 명세단계에서 문법적 오류가 존재하는지를 검사하는 것이다. 정확성에 대한 검증은 'Z formalizer'와 같은 도구를 이용하여 자동으로 검사할 수 있다. 그리고 일관성과 완전성은 다음과 같이 증명한다.

- 일관성

앞서 명세 된 보안정책모델은 재귀적 데이터타입이나 불변식(invariant)을 가진 데이터타입을 포함하지 않

는다. 따라서 초기상태가 존재함을 보이면 일관성에 대해 증명할 수 있다. 이것을 정형적으로 표현하면 아래와 같다.

$InitExist \triangleq \exists Scheduler' \bullet SchedulerInit$

*InitExist*

$\exists current' : OptPId;$
$ready' : \mathbb{P}\, PId;$
$blocked' : \mathbb{P}\, PId;$
$free' : \mathbb{P}\, PId;$
$(\{current'\} \backslash \{nullPId\},$
$ready',$
$clocked',$
$free')$ partition $PId \bullet$
$current' = nullPId$
$ready' = \emptyset$
$blocked' = \emptyset$
$free' = PId$

one-point rule(아래 설명되어 있음)을 네 번 적용하여 존재한정사(existential quantifier)를 제거하고, 술어 축약을 수행하면, 'InitExist'는 다음과 같이 간략해 진다.

*InitExist*

$true$

초기 상태가 존재함을 의미하는 'InitExist'의 내부 서술이 모두 참이 되어, 평가대상인 스케줄러의 보안 정책모델의 일관성을 증명할 수 있었다.

참고) one-point rule

$$\exists x : X \bullet P(x) \wedge x = t \vdash t \in X \wedge P[t \backslash x]$$

- 완전성

전제조건(Precondition)에 대한 일반적인 형태는 아래와 같이 표현될 수 있다.

$$preOperation = \exists State' \bullet Operation \backslash outputs$$

여기서, 시스템의 상태는 'State' 스키마에 의해 모델링 되고, 'outputs'는 수행과 연관된 출력들의 목록이다.

생성(Create) 기능을 위한 전제조건은 다음 스키마로 주어진다.

*Scheduler*
$g? : PROGRAM$
$p! : PId$

$\exists Scheduler' \bullet$
$current' = current$
$ready' = ready \cup \{p!\}$
$blocked' = blocked$
$free' = free \backslash \{p!\}$
$p! \in free$

앞서 증명한 일관성 증명과 같은 방법을 이용하면 스키마 내부의 모든 서술이 참이 됨을 보일 수 있다. 이로서 'PreCreate'가 'Create'의 전제조건이 되며, 'Create'가 완전함을 증명할 수 있다.

## Ⅶ. 결 론

현대사회에서 정보보호의 중요성은 항상 강조되고 있는 부분이며, 그 중요도는 지속적으로 증가하고 있는 추세이다. 정보의 중요도가 높아질수록 요구되는 정보보호 수준은 차이가 발생하게 된다. 모든 정보가 디지털화되어 관리되고 있는 현실에서 통신상에서 정보보호 기술이 향후 중요한 국가 기술력의 하나이며, 이러한 이유로 보안 기능을 신뢰할 수 있는 정보보호 제품을 개발하고 평가하는 것은 중요한 국가 기술력의 자리 잡을 것이라 판단된다.

본 고의 도입부에서는 정보보호 제품을 평가 기준인 공통평가기준과 그 안에서 고 신뢰도의 제품을 개발, 평가할 수 있는 정형기법의 상호관계와 국외 동향에 대해서 기술하며, 신뢰할 수 있는 보안 기능을 위해서는 정형기법을 통해서 일관성 있게 계획, 명세 및 개발 검증이 이루어져야만 함을 살펴보았다. 아울러, 정보의 중요도와 그 정보를 보호하기 위한 국가 보안 정책에 의해서 고등급 정보보호 제품의 필요성이 증대된다는 점을 다시 확인할 수 있었다.

중반부에서는 정형기법에 대한 간단한 소개와 개발 클래스의 근간이 되는 보안정책모델에 대해 정형명세언어인 Z를 사용하여 명세 검증하는 방법에 대한 소개를 통해서 고등급 평가를 위해서 제품 명세 및 검증이 진행되는 과정을 기술하였다. 이를 통해서 시스템의 일관성과 완전성을 확인하는 방법에 대해서도 살펴보았다.

현재, 평가적체가 발생할 정도로 정보보호 산업 전반에 대한 관심과 정보보호제품 개발이 증가하고 있는 시점에서 향후 국가기간망에 대한 보다 안전한 정보보호와 신뢰도가 높은 정보보호제품 개발을 위하여 고등급 평가와 관련된 기술에 대한 지속적인 연구와 관심이 필요하다고 판단된다.

## 참고문헌


[1] "Common Criteria for Information Technology Security Evaluation", Ver 3.1, September 2006, CCMB-2006-09-003

[2] Frank Koob, Markus Ullmann, Stefan Wittmann, "The New Topicality of Using Formal Models of Security Policy within the Security Engineering Process", Lecture Note in Computer Science, Springer, LNCS 1641, pp. 302~310, 1998

[3] R. Richards, D. Greve, M. Wilding, W. Mark Vanfleet, "The Common Criteria, Formal Methods, and ACL2," ACL2 Workshop 2004.

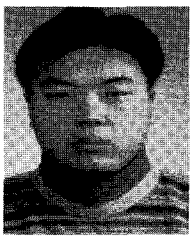
[4] Mark S. Merkow and Jim Breihaupt, Computer security assurance using the common criteria, Thomson/Delmar Learning, Clifton Park, NY, 2005

[5] Woodcook, J. Davies, J., "Using Z : Specification, refinement, and proof" Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1996

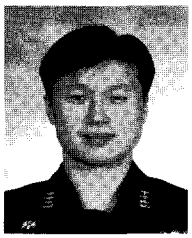
[6] Information Technology Security Evaluation Criteria, Office for Official Publications of the European Communities, Lexembourg 1991.

[7] Junkil Park, Jin-Young Choi,"Formal Security Policy Model for a Common Criteria Evaluation", ICACT 2007, Korea, February 2007, pp. 277-281

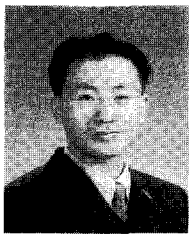

## 〈著者紹介〉

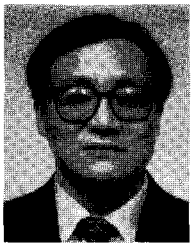
**심 재 환 (Jae-Hwan Sim)**
2002년 8월 : 연세대학교 기계공학부 졸업
2004년 2월 : 고려대학교 컴퓨터학과 석사 졸업
2006년 3월 ~ 현재 : 고려대학교 컴퓨터·통신공학부 박사과정
<관심분야> 실시간시스템, 임베디드 시스템



**김 현 석 (Hyun-Seok Kim)**
2000년 2월 : 육군사관학교 관리학과 졸업
2004년 2월 : 고려대학교 컴퓨터학과 석사졸업
2006년 3월 ~ 현재 : 고려대학교 컴퓨터 통신공학부 박사과정
<관심분야> 무선네트워크, 보안프로토콜, 전자상거래



**유 희 준 (Hee-Jun Yoo)**
1997년 8월 : 고려대학교 컴퓨터학과 이학학사
1999년 8월 : 고려대학교 컴퓨터학과 이학석사
2005년 2월 : 고려대학교 컴퓨터학과 이학박사
2005년 3월 ~ 2007년 1월 : 삼성전자 TN총괄 무선사업부 개발1그룹 책임연구원
2007년 2월 ~ 현재 : 한국정보보호진흥원 보안성평가단 평가서비스팀 선임연구원
<관심분야> 정형명세언어 Z, 정리증명, 규칙 기반 시스템, 임베디드 시스템, 고등급 평가, 취약성 분석 평가



**최 진 영 (Jin-Young Choi)**
정회원
1982년 2월 : 서울대학교 컴퓨터공학과 졸업
1986년 2월 : Dept. of Mathematics and Computer Science, Drexel University 석사
1993년 3월 : Dept. of Computer and Information Science, University of Pennsylvania 박사
1996년 ~ 현재 : 고려대학교 컴퓨터·통신공학부 교수
<관심분야> 정형기법, 임베디드 실시간시스템, 프로그래밍언어, 프로세스 대수, 소프트웨어 공학