공통평가기준 v3.1 기반 고등급 평가방법론

노 병 규* 유 희 준*

목 차

1. 서 론

정보통신 기술과 인터넷 기반 기술과 인프라가 급속도로 발전하면서 인터넷은 인간의 생활에 큰 영향을 미치게 되었다. 과거 실생활에서 이루어지던 많은 활동들이 인터넷에서도 가능하게 되면서, 많은 사람들이 사용의 편의성 등의 이유로 활용 분야 및 빈도가 증가하고 있는 추세이다. 이런 이유로 인터넷 상에는 보다 많은 정보와 서비스가 제공되고 있으며, 이러한 정보와 서비스를 사용하기 위해서 개인의 중요한 정보를 요구하는 경우가 빈번하게 발생하게 되었다. 하지만, 이러한 정보들은 사용자들의 편리하게 정보를 획득하게 하기 위해서 제공되는 정보 검색 및 정보 공유에 관련된 기술들에 의해서 정보를 쉽게 획득하고 가공할 수 있도록 해주었다. 이로 인하여, 정보 획득과 제공이라는 순기능과 함께 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(CC: Common Criteria)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. 또한, 올해부터는 평가에서 적용되는 공통평가기준의 버전이 v2.3에서 v3.1로 변경되었다. [1][2]

따라서 신규 버전에 대한 새로운 평가방법론이 필요한 시점이 되었다. 본 고에서는 신규 적용되는 CC v3.1과 이전 버전인 v2.3을 고등급 평가 관점에서 비교한 후, v3.1 기반 고등급 평가 방법론을 제시하겠다. 본고의 구성은 다음과 같다.

2장에서는 공통평가기준과 공통평가기준에서 보증등급에 따른 정형화 정도 및 정형화 될 문서에 대한 언급과 정형기법에 관하여 살펴볼 것이다. 3장에서는 지금까지 사용된 CC v2.3과 신규 버전인 v3.1사이의 고등급 평가 시의 차이점을 살펴본 후에, 4장에서는 새로 적용되는 v3.1에서 고등급 평가 방법을 제시한 후, 결론을 맺도록 하겠다.

* 한국정보보호진흥원 보안성평가단

2. 공통평가기준과 정형기법

2.1 공통평가기준

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(EAL: Evaluation Assurance Level)을 정의하고 있다. 특히 EAL5에서 EAL7까지는 고 보증등급으로 불린다. [1][3][4]

CC 기반의 평가·인증 제도를 사용하고 있는 국가들은 서로의 평가·인증 결과를 상호 인정해주기 위해서 공통평가기준 상호인정협약(CCRA: Common Criteria Recognition Arrangement)를 맺고 있다. CCRA에는 인증서를 발행하는 국가들(CAP: Certificate Authorizing Participants)과 인증서를 수용하고 있는 국가들(CCP: 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(PP: Protection Profile)이라는 개념으로 소개한다. 평가대상(TOE: Target of Evaluation)은 평가의 대상이 되는 전체 혹은 부분 시스템을 의미하는 개념이다. 보안목표(ST: Security Target)는 평가자가 평가의 기반으로 사용할 정보로서 평가대상에 대해 보안목표가 만족됨을 보이는 인증과 이에 관련된 문서를 결과로 산출한다. 공통평가기준의 평가보증등급은 표 1과 같다.

(표 1) 공통평가기준의 평가보증등급

EAL7	정형적으로 검증된 설계 및 시험
EAL6	준 정형적으로 검증된 설계 및 시험
EAL5	준 정형적인 설계 및 시험
EAL4	조직적인 설계, 시험 및 검사
EAL3	조직적인 시험 및 검사
EAL2	구조적인 시험
EAL1	기능적인 시험

표 1에서 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(ADV: Development), 형상관리(ACM: Configuration Management), 시험(ATE: Testing) 등의 클래스로 분류되어있다. 이 중 실제 보안시스템의 개발 단계에서 작성되고 정형기법과 관련된 문서는 개발 클래스이다.

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, 이들간의 일치성을 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. 이러한 개발클래스는 v2.3과 v3.1사이에 다소 차이점을 가지고 있으며, 이와 관련된 자세한 사항은 3장에서 다루도록 하겠다.

2.2 정형기법

하드웨어나 소프트웨어 시스템은 불가피하게 점점 더 규모나 기능면에서 커지고 복잡해지고 있는 실정이다. 이러한 복잡한 시스템에 대해서 과거와 같이 테스트 혹은 시뮬레이션만으로는 작은 에러를 발견하기가 어렵다.[5]

이러한 문제를 해결하기 위해서 최근에 와서는 시스템 개발에 정형기법이 사용되고 있다. 정형기법은 수학과 논리학에 기반을 둔 방법으로 하드웨어나 소프트웨어 시스템을 명세하거나 검증하는 것이다. 수학적 기호를 사용하여 시스템이 특성을 만족하는지를 수학적 성질을 이용하여 검증하므로 자연어가 내포하고 있는 불확실성을 최소화할 수 있으며, 따라서 복잡한 시스템에서 시스템이 만족해야만 하는 특성 혹은, 반드

시 만족하지 않아야 하는 특성에 대한 수학적 검증을 수행할 수 있다. 이러한 정형기법(Formal Methods)은 정형 명세(Formal Specification)와 정형 검증(Formal Verification)으로 구분된다.

정형 명세 언어는 자연어가 내포하고 있는 애매모호함이 없기 때문에 명세하고자 하는 시스템을 명확하게 기술할 수 있다. 각각의 정형 명세 언어마다 제공하는 도구들을 사용하여 명세가 오류 없이 명확하게 명세 되었는지를 검사하여 명세 시 발생할 수 있는 오류를 제거해준다. 정형명세언어에는 집합론, 논리에 기반을 한 Z, VDM과 상태를 기반으로 하는 StateChart, 마지막으로 CCS, CSP, ACSR과 같은 프로세스 대수를 기반한 언어들이 있다.

정형 검증 방법에는 크게 모델체킹과 정리증명 방법이 있으며, 각각에 대해서 간단히 살펴보면 다음과 같다.

먼저, 모델체킹은 시스템에 대한 유한 상태 모델과 검사하고자 하는 속성이 논리적 정형성을 가지고 만들어졌을 때, 이 시스템이 해당 속성을 만족하는지를 자동적으로 검사하는 방법이다. 이 방법은 상태 탐색(State Exploration)을 기반으로 하며, 상태 전이 시스템(State Transition System)과 속성이 주어지면, 주어진 시스템이 검증하고자 하는 속성을 만족하는 지를 알아보기 위해서 전체 상태를 검사하는 방법이다.

이 방법의 장점은 자동화된 검증도구가 지원된다는 것이다. 사용자가 시스템의 모델을 입력하고 요구사항 명세를 나타내는 속성들을 입력하면 도구는 자동적으로 모델의 상태를 검사하여, 속성을 만족하는지 여부를 검증하여, 만족하지 못하는 경우에는 반례를 보여주어 모델의 어느 부분이 잘못되었는지 여부를 쉽게 확인할 수 있도록 도와준다. 하지만, 이 방법은 모델의 상태의 수가 커질 경우에는 상태폭발문제로 인해, 검증할 수 없게 되는 단점이 있다.

정리 증명은 수학적인 논리식을 사용하여 시스템과 시스템 상에서 요구되는 특성들을 표현하는 기술이며, 주어진 시스템의 공리로부터 특성에 대한 증명을 사용하여 논리식의 진위를 발견해 내는 과정이다. 증명의 각 단계들은 공리와 규칙에 의존하고 파생된 정의들을 증명해 나가는 방법을 사용하여 논리식을 증명해 나가는 방법이다. 이 방법의 장점은 모델 체킹의 단점인 무한상태에 대해서 직접적으로 다룰 수 있다는 점이다. 무한 공간 집합에 대한 증명을 하기 위해서 귀납법등과 같은 증명 기법을 사용하기 때문에 모델 체킹에서 발생하는 상태 폭발 문제를 해결할 수 있다. 하지만, 규칙을 적용하는 부분에서는 도구의 도움을 받을 수 있어도 정의 부분에서 상호작용이 필요하기 때문에 숙련된 논리전문가의 도움이 필요하다는 문제점이 있다.

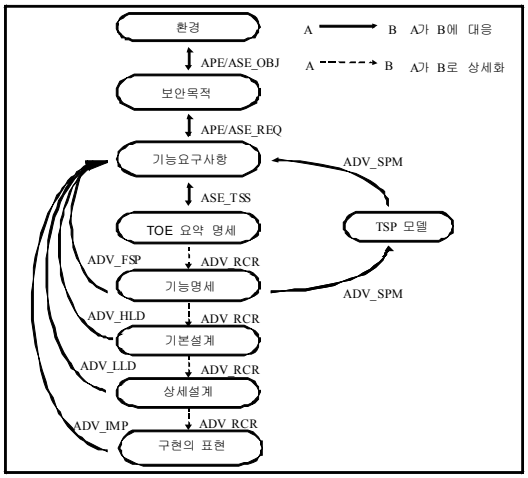
3. CC v2.3 과 v3.1의 차이점

공통평가기준은 올해부터 평가의 기준 버전이 v2.3에서 v3.1로 변경되었다. CC v3.1은 불필요하거나 효과가 낮은 업무 배제, 명확한 용어 정의, 평가 업무 재구성 및 새로운 요구사항 추가를 편하게 함을 목적으로 하고 있다. 두 버전은 여러 클래스 상에서 다소 차이점을 보이고 있다. 우선, v2.3에서 독립적으로 평가를 수행했던 형상관리 클래스(ACM Class)와 배포 및 운영 클래스(ADO Class)는 v3.1에서 삭제되면서, 설명서 클래스(AGD Class)와 생명주기 클래스(ALC Class)의 연관이 있는 부분과 결합해서 평가하도록 구성되어 있다. 본 고에서는 주로 살펴보려는 개발 클래스와 같은 경우에도 보증 패밀리 상에 다소 변화가 있다.

3.1 개발클래스 상의 차이점

개발문서의 변경사항을 살펴보면, 표현의 일치성(RCR: Representation correspondence) 문서가 v3.1에서 요구되지 않으며, 상세설계서(LLD: Low-level Design)와 기본설계서(HLD: High-level Design)로 구분되었던 설계문서가 TOE 설계서(TDS: TOE design)로 통합되었다. 전체적으로 TDS 패밀리가 이전의 두 패밀리에 비해 TOE 구성 및 설계에 대한 보다 높은 수준의 분석과 설명을 요구하고 있으며, TOE 보안구조(ARC: Security Architecture)이 추가되었다. ARC에서는 자체보호, 우회불가성, 영역분리 및 TSF 초기화의 안정성에 대해 서술하며, 이는 보안기능요구사항(SFR: Security Function Requirements)-수행 설명과 동일한 상세수준으로 제공되어야 한다. 자체보호와 우회불가성의 경우, 보안기능이 정확히 구현되었는지를 검증하는 것이 어려우며 오

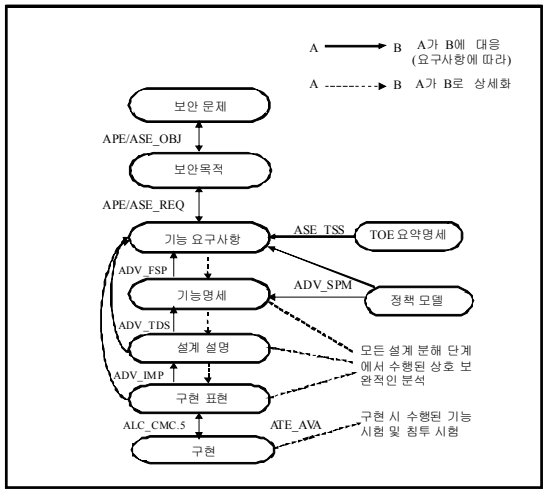
히려 보안기능 측면보다는 TOE 설계의 정확한 구현을 통해 수행되는 TOE 보안기능(TSF: TOE Security Functions) 특성이 있으므로, 보안기능 요구사항에서 보증요구사항으로 이동되었다.



(그림 1) TOE 표현과 요구사항 간의 관계도

그림 1은 v2.3에서 TOE 표현과 요구사항들에 대한 관계를 표현하고 있다. 그림 1은 다양한 TSF 표현과 그들이 다루고자 하는 목적 및 요구사항 간의 관계를 나타낸다. 그림과 같이 보호프로파일 평가 클래스(APE)와 보안목표명세서 평가 클래스(ASE)는 TOE의 예상되는 환경과 보안목적간의 일치성 및 보안 목적과 기능요구사항간의 일치성에 대한 요구사항을 정의한다. 보안목표명세서 평가 클래스는 보안목적 및 기능요구사항과 TOE 요약명세간의 일치성에 대한 요구사항도 정의한다.

위에서 언급한 것을 제외한 그림 1에서 나타나는 모든 일치성에 대한 요구사항은 개발클래스에서 정의된다. 보안정책모델(SPM: Security Policy Model) 패밀리는 TOE 보안정책과 TOE 보안정책모델간의 일치성 및 TOE 보안정책과 기능명세간의 일치성을 정의한다. 표현의 일치성 패밀리는 TOE 요약명세에서 구현의 표현까지의 모든 이용 가능한 TSF 표현간의 일치성에 대한 요구사항을 정의한다. 마지막으로, TSF 표현에 관한 각 보증패밀리는 기능요구사항에 대해 해당 TSF 표현과 관련된 요구사항을 정의한다. 이들의 조합은 TOE 보안기능요구사항이 다루어지고 있음을 보장하는데 도움을 준다. 추적가능성 분석은 가장 기본 단계의 TSF 표현으로부터 제공되는 모든 TSF 표현의 상세화 단계에 대하여 수행되어야 한다. 공통평가기준은 추적가능성에 대한 요구사항을 표현의 일치성 패밀리(ADV_RCR)에 대한 종속관계를 통해 요구한다. TSF 내부 패밀리(ADV_INT)는 TSF의 내부구조와 관련이 있기 때문에 그림에는 나타나지 않으며, TSF 표현의 상세화 과정에만 간접적으로 관련이 있다.



(그림 2) 개발클래스의 패밀리 간 관계 및 다른 클래스와의 관계

반면, 그림 2는 v3.1에서 개발 클래스 내의 다양한 TSF 표현 간의 관계 및 다른 클래스와의 관계를 나타낸다. 그림에서 알 수 있듯이 보호프로파일 평가(APE) 클래스와 보안목표명세서 평가(ASE) 클래스는 보안기능요구사항(SFR)과 TOE 보안목적 간의 일치성에 대한 요구사항을 정의한다. 또한, ASE는 보안목적과 SFR 간의 일치성에 대한 요구사항 및 TOE가 SFR을 만족시키는 방법에 대해 서술하는 TOE 요약명세에 대한 요구사항을 정의한다. ALC_CMC.5.2E 평가활동은 시험(ATE) 클래스와 취약성 평가(AVA) 클래스에서 시험된 TSF가 개발(ADV) 클래스의 분해 단계에서 모두 서술된 사항임을 검증한다.

그림 2에 나타난 일치성에 대한 요구사항은 개발 클래스에 정의되어 있다. 보안정책모델(ADV_SPM) 패

밀리는 선택된 SFR을 정형화하여 모델링하고, 기능명세와 정형화된 모델간의 일치성 제공에 대한 요구사항을 정의한다. TSF 표현과 관련된 각 보증 패밀리(즉, 기능명세(ADV_FSP), TOE 설계(ADV_TDS), 구현의 표현(ADV_IMP))는 해당 TSF 표현을 SFR로 대응시키는 요구사항을 정의한다. 모든 분해는 다른 분해들을 정확하게 반영해야 한다(예: 모든 분해는 상호 보완적임). 개발자는 각 컴포넌트의 마지막 증거 요구사항(.C)에서 추적에 대한 정보를 제공한다. 이와 관련된 보증은 특정 수준의 분해에 대한 분석을 수행하는 동안, 다른 수준의 분해(예: 순환 방식)를 참조하여 각 수준의 분해를 분석함으로써 얻게 된다. 평가자는 두 번째 평가자 요구사항(.E)의 일부로써 일치성을 검증한다. 이러한 분해 수준으로부터 얻는 지식은 기능 시험 및 침투 시험 노력의 기초가 된다.

ADV_INT는 TSF 내부 구조와 관련된 것으로 TSF 표현의 상세화 과정에만 간접적으로 연관되기 때문에 그림 2에 포함되어 있지 않다. 이와 유사하게, ADV_ARC는 TSF 표현보다는 구조적 적절성과 관련된 요구사항이므로 그림 2에 표현되지 않았다. ADV_INT와 ADV_ARC는 TOE 보안기능성이 우회되거나 손상되지 않는다는 특성과 관련된다.

앞서 살펴본 것과 같이 v2.3과 v3.1은 개발클래스 패밀리의 종류와 그들간의 상호작용에서 다소 차이를 보이고 있는 것을 알 수 있었다. 다음 절에서는 본 고에서 설명하고자 하는 고등급 평가의 경우 나타나는 차이점에 대해서 설명하도록 하겠다.

3.2 고등급 평가 시 두 버전의 차이점

지금부터는 고등급 평가 시에 이들 개발 보증 클래스 상에 어떠한 차이점이 있는지를 살펴보도록 하겠다. 우선, 고등급 평가를 위해서는 개발문서에 대한 제출물은 정형 혹은 준정형 방법론을 적용하여 작성되어야만 한다. 목표하는 등급에 따라서 정형방법론을 적용할 것인지 준정형 방법론을 적용할 것인지 결정하게 되며, 정형방법론을 사용하는 경우에 더 높은 등급의 평가를 받을 수 있게 된다. 따라서, 가장 높은 등급인 EAL7에서는 대부분의 개발문서들이 정형방법론이 적용되어 작성되어야만 한다.

이러한 정형과 준정형 방법론의 가장 큰 차이는 방법론이 명확한 의미론(Semantics)를 가지고 있는지 여부의 차이이다. 방법론이 의미론을 가지고 있으면 해당 방법론으로 작성된 모델에 대한 일관성 및 완전성을 검증할 수 있다는 의미이며, 이러한 수학적 증명은 고등급 평가에서는 반드시 필요한 부분이다.

표 2는 v2.3에서 개발클래스의 정형방법론에 대한 보증요구사항을 나타내고 있다. 내용을 살펴보면, 고등급 제출물 작성에 정형방법론을 적용해야 하는 클래스는 기능명세(FSP: Functional Specification), 기본설계(HLD), 상세설계(LLD), 표현의 일치성(RCR)과 보안정책모델(SPM)이며, SPM의 경우는 유일하게 EAL5 이상에 모두 정형 방법론을 적용해야만 한다. 반면 LLD를 제외한 나머지 제출물들은 EAL7에서만 정형방법론을 적용해야 한다. 그 외의 나머지 제출물들은 준정형 방법론을 적용하여 기술하면 된다.

(표 2) v2.3에서 개발문서 정형기법 요구사항

클래스	패밀리	평가 보증 등급		
		EAL5	EAL6	EAL7
ADV	FSP	SF	SF	F
	HLD	SF	SF	F
	IMP	I	I	I
	INT	I	I	I
	LLD	I	SF	SF
	RCR	SF	SF	F
	SPM	F	F	F

F : Formal SF : Semi Formal I : Informal

표 3은 v3.1에서 정형 방법론에 대한 요구사항을 기술하고 있다. 정형 방법론이 필요한 문서는 FSP, SPM과 TDS이다. 이전 버전과 비교해 보면, FSP의 경우 보증요구사항의 큰 변화가 없지만, SPM의 경우는 이전 버전과 다르게 EAL5 버전에서는 SPM 문서가 요구되지 않는다. 이는 v3.1에서는 EAL5등급 평가가 이전 버전에 비해 정형기법 요구사항이 다소 낮아진 것을 의미한다고 볼 수 있다. 반면, HLD와 LLD가 합쳐진 TDS의 경우는 EAL7 등급 평가에 있어서 정형 방법론이 적용되어야 함으로 이전 버전에 비해 좀 더 정형기법 요구사항이 높아졌다고 볼 수 있겠다.

(표 3) v3.1에서 개발문서 정형기법 요구사항

클래스	패밀리	평가 보증 등급		
		EAL5	EAL6	EAL7
ADV	ARC	I	I	I
	FSP	SF	SF	F
	IMP	I	I	I
	INT	I	I	I
	SPM	N/A	F	F
	TDS	SF	SF	F

F : Formal SF : Semi Formal I : Informal

SPM의 경우 v2.3에서는 EAL4등급에서 비정형 모델을 기술하는 경우가 있었지만, v3.1에서는 SPM에 대해 정형적인 방법론으로만 모델을 기술하도록 요구하고 있다. 이는 TOE의 정책 모델을 규칙의 집합으로 기술하던 방식에서 보다 추상적인 형태의 모델이 요구된다고 생각할 수 있겠다. 개인적인 견해로는 새로운 버전에서는 보다 쉽게 EAL5 등급을 접하게 되며, EAL7의 경우는 보다 엄격해졌다고 생각된다.

4. 고등급 평가 방법론

앞서 보았듯이 CC v2.3과 v3.1은 많은 차이점을 가지고 있는 이유로, 새로운 버전에서 고등급 평가를 위해서는 이전 버전과 다른 방법론이 적용되어야만 하며, 그에 맞는 새로운 평가 기술 및 방법론을 개발이 필요하며, 이를 위해 우리는 신규 버전에 대한 고등급 평가를 위한 새로운 방법론을 제시하고자 한다. 본 고에서는 대표적인 고등급 제출물인 SPM(Security Policy Model)에 대한 평가방법론을 제시하고자 한다.

SPM은 개발클래스 중에서 유일하게 정형방법론 적용만을 요구하는 패밀리이다. 이 패밀리의 목적은 TSF에 대한 정형화된 보안정책모델을 개발함으로써 추가적인 보증을 제공하고, 기능명세와 보안정책모델 간의 일치성을 확립한다. 보안정책모델은 수학적인 증거에 의해 그 특성으로부터 정형화된 보안 원칙을 수립하는 것이다. SPM 평가에서 차이가 발생하는 이유를 해당 패밀리에 대한 두 버전의 차이점을 개발자와 평가자 관점에서 보다 상세하게 비교하면서 설명하도록 하겠다. 본 고에서 설명하고 있는 평가방법론은 CC v3.1 기반의 EAL6등급을 대상으로 하고 있다. 그 이유는 표 3에 표시된 것과 같이 EAL5의 경우 SPM 문서가 요구되지 않으며, EAL7등급의 경우는 대부분의 제출물이 정형방법론을 적용하여 개발되기 때문에 각 보증 패밀리간의 일치성과 완전성을 수학적으로 증명할 수 있기 때문이다. 반면에 EAL6의 경우, SPM만 정형방법론이 적용되며, 나머지 패밀리에 대해서는 준정형 방법론이 적용되는 이유로 EAL7과 비교해서 수학적인 증명을 이끌어 내기가 어렵다고 생각된다.

4.1 개발자 요구사항에서 차이점

본 절에서 비교가 되는 패밀리는 정형방법론적용이 요구되는 v2.3의 ADV_SPM.3와 v3.1의 ADV_SPM.1이다. 다음은 v2.3의 ADV_SPM.3 패밀리에 대한 개발자 요구사항이다.

ADV_SPM.3.1.D	개발자는 TSP 모델을 제공해야 한다.
ADV_SPM.3.2.D	개발자는 기능명세와 TSP 모델간의 일치성을 입증하거나, 적절하게 증명해야 한다.

반면에, v3.1에서는 보다 명확한 모델과 관련 문서와의 일치성을 제공할 것을 요구하고 있다. 다음은 v3.1의 ADV_SPM.1 패밀리에 대한 개발자 요구사항이다.

ADV_SPM.1.1.D	개발자는 [할당: 정형화된 방식으로 모델링 된 정책목록]에 대한 정형화된 보안 정책 모델을 제공 해야 한다.
ADV_SPM.1.2.D	정형화된 보안 정책 모델에서 다루는 각 정책에 대하여, 그 모델은 정책을 구성하는 SFR의 관련 부분을 식별해야 한다.
ADV_SPM.1.3.D	개발자는 모델과 정형화된 기능 명세간의 일치성에 대한 정형화된 증거를 제공해야 한다.
ADV_SPM.1.4.D	개발자는 모델과 기능명세 간의 일치성 입증을 제공해야 한다.

이전 버전에서 기술된 두 가지로 기술되었던 개발자 요구사항이 보다 명확하게 기술되었다는 것을 확인

할 수 있다. 제공해야 되는 보안정책모델이 정의된 보안기능요구사항에서 식별된 정책을 모델링하고 정형화된 기능명세 유무에 따라서 정형화된 증거 혹은 일치성 입증을 제공하도록 요구하고 있다. 따라서, 표 3에서 확인할 수 있듯이 개발자는 정형 FSP가 제공되는 EAL7에는 ADV_SPM.1.3.D를 제공할 수 있지만, 그렇지 않은 EAL6에서는 ADV_SPM.1.4.D를 제공하게 될 것이다. 이런 이유에서 새로운 평가 방법론을 제시하게 되었다. 평가자는 위와 같은 요구사항에 적합하도록 제출된 문서를 가지고 다음에 기술되는 증거요구사항을 가지고 제공된 정보가 모든 증거요구사항을 만족하는지 확인해야 한다.

4.2 증거 요구사항에서 차이점

다음은 v2.3에서 개발자가 제출하는 SPM문서에서 증거로 제출되어야 하는 요구사항이다.

ADV_SPM.3.1.C	TSP 모델은 정형화되어야 한다.
ADV_SPM.3.2.C	TSP 모델은 모델링 가능한 TSP의 모든 정책의 규칙과 특성을 서술해야 한다.
ADV_SPM.3.3.C	TSP 모델은 모델링 가능한 TSP의 모든 정책이 일관성 있고 완전함을 보이는 이론적 근거를 포함해야 한다.
ADV_SPM.3.4.C	TSP 모델과 기능명세간의 일치성을 보일 경우에는 기능명세에 명시된 모든 보안기능이 TSP 모델에 대하여 일관성 있고 완전한지 입증해야 한다.
ADV_SPM.3.5.C	기능명세가 준정형화되어 있는 경우, TSP 모델과 기능명세간의 일치성 입증은 준정형화 되어야 한다.
ADV_SPM.3.6.C	기능명세가 정형화되어 있는 경우, TSP 모델과 기능명세간의 일치성 증명은 정형화되어야 한다.

반면, v3.1 정책모델을 설정한 요인들을 식별하고, TOE가 안전한 상태에서 일관성있고 완전한지를 이전 버전보다 좀 더 명확하고 상세하게 기술하도록 요구하고 있다. SPM의 관련문서인 FSP와 일관성 있게 기술되었는지를 입증하고 확인이 필요하다.

ADV_SPM.1.1.C	모델은 요구되는 경우 설명문이 지원되는 정형화된 방식이어야 하며, 모델링된 TSF의 보안정책을 식별해야 한다.
ADV_SPM.1.2.C	모델링된 모든 정책에 대해 모델은 TOE에 대한 보안성을 정의하고, TOE가 안전하지 않은 상태가 될 수 없다는 정형화된 증거를 제공해야 한다.
ADV_SPM.1.3.C	모델과 기능명세 간의 일치성은 정확한 정형화 수준이어야 한다.
ADV_SPM.1.4.C	일치성은 기능명세가 모델에 대하여 일관성 있고 완전함을 보여야 한다.
ADV_SPM.1.5.C	일치성 입증은 기능명세 내의 인터페이스가 ADV_SPM.1.1D에서 할당된 정책에 대하여 일관성 있고 완전하다는 것을 보여야 한다.

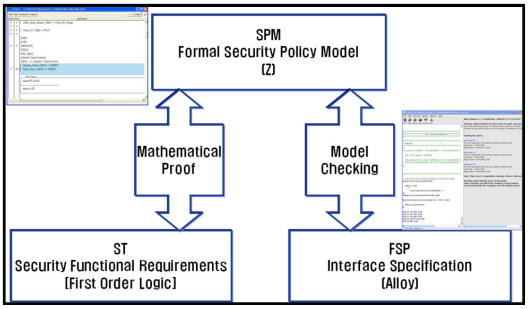
앞에서도 언급했듯이, 준정형 방법론으로 작성된 FSP가 제공되는 EAL6의 경우는 개발자가 일관성과 완전함을 정형적인 방법으로 증명하기가 매우 어렵기 때문에, 개발자는 증명보다는 쉬운 예제를 통한 입증을 제공할 것으로 생각된다. 하지만, 평가자는 고등급 평가에 있어서는 모든 요구사항이 완전하고 일관성있게 기술되어 있는지를 정형적인 방법으로 확인하는 것이 반드시 필요하다. 이 문제를 해결하기 위해서 다음 절에 제안된 방법론을 사용하였다.

4.3 제안하는 SPM 평가방법론

우리는 v3.1기반의 고등급 평가 방법론 개발을 위해서 우선 스마트카드 전자지갑 응용프로그램인 K-Debit 카드를 대상으로 EAL6등급 평가 제출물을 작성하며, 그에 대한 평가 작업을 수행하고 있는 중이며, 본 고에서 제안되는 방법도 현재 수행중인 평가의 일부 결과물이다. 대상 TOE는 K-Debit 카드 응용프로그램과 운영체제 및 그들이 구동되는 IC칩을 포함한다. SPM 문서는 정형방법론인 Z[6]를 사용하여 작성되었으며, FSP는 준정형방법론인 UML[7]의 Sequence Diagram등으로 작성되었다. 해당 제출물을 가지고 앞으로 소개할 방법론을 사용하여 평가를 수행하고 있다.

SPM 문서 평가를 위해서는 관련된 제출물인 보안목표명세서(ST: Security Target)과 기능명세서(FSP)사이의 일관성을 확인하는 것이 매우 중요하다. 우선, 우리

가 일관성을 확인하기 위해서 제안한 방법론을 그림 3과 같이 요약 표시하였다.

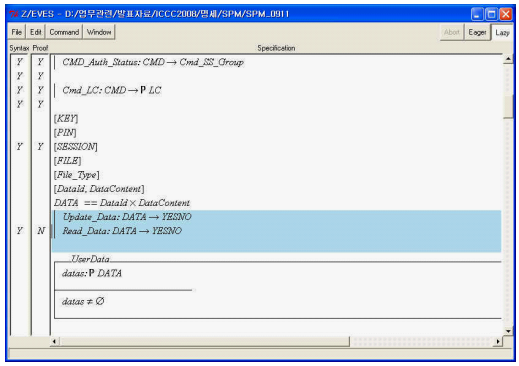


(그림 3) 제안된 SPM 평가방법론 요약

ST와 일관성을 확인하기 위해서, 모델을 정의하는데 사용된 보안기능 요구사항을 논리식으로 변환한 후에, 제공된 Z로 작성된 SPM과의 일치성을 정형 도구인 Z/EVE[8]를 사용하여 수학적 증명을 수행하였으며, FSP와의 일관성을 위해서는 Z와 유사한 표현력을 가지고 있는 모델 체킹 방법인 Alloy[9]로 모델을 변환하고, FSP의 인터페이스를 Alloy로 추가 명세한 후에 모델 체킹 방법을 사용하여 일관성을 확인하였다. 다음은 우리가 사용한 방법을 순차적으로 나열한 것이다.

1. TOE의 보안요구사항들을 참고하여 정형 보안정책모델을 만든다.
2. 정형 보안정책모델에 대한 내부 일관성과 완전성을 검증한다.
3. 보안 요구사항을 논리식으로 변환한다.
4. 변환된 논리식과 정형 보안정책모델을 정형도구를 사용하여 수학적 증명을 수행한다.
5. TSF 인터페이스 모델을 효과와 예외사항 등을 기준으로 만든다.
6. 보안정책모델과 TSF 인터페이스 모델 사이의 모델 체킹을 수행한다.

Z와 Alloy는 동일하게 논리기반의 방법론으로 의미가 유사하며, Alloy가 Z 명세에 대한 모델 체킹을 목적으로 개발된 배경을 갖고 있기 때문에 두 모델 사이의 변환이 다른 경우들에 비해 쉽게 수행할 수 있다는 장점을 가지고 있다. 따라서, 우리가 제안한 방법론은 SPM이 Z 방법론을 사용하여 작성하였을 경우 가장 효과적으로 적용할 수 있으며, Z가 아닌 다른 방법론을 사용한 경우에는 가장 유사한 기반 이론을 가진 모델 체킹 방법론을 찾아 사용하는 것이 효과적이라고 할 수 있겠다. 또한, ST의 보안기능 요구사항은 주체, 객체의 행동을 정의하고 있어 논리식으로 쉽게 변환 가능하며, 논리기반 언어인 Z의 지원도구인 Z/EVES를 사용하여 보안정책모델과 논리식 간의 수학적인 증명을 통해서 만족 여부를 검증할 수 있다. 그림 4는 Z로 작성된 TOE에 대한 보안정책 모델의 일부분이다.



(그림 4) TOE의 보안정책모델 (일부분)

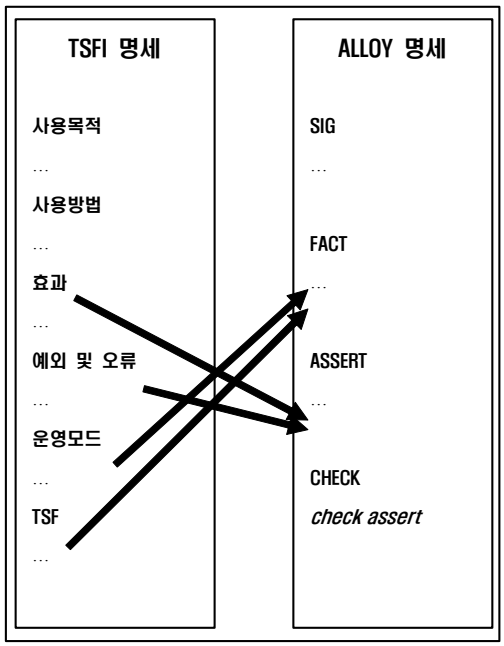
Z/EVE를 사용하여 작성된 보안정책모델에 대한 내부 일관성과 완전성을 검증한 후, 각각의 명세된 서브 보안기능을 최종적으로 통합한 TOE에 대한 Z 스키마에 논리식으로 변환한 보안기능 요구사항을 정리(Theorem) 형태로 만들어서 증명을 수행하였다. 다음은 최종 스키마와 Key block 보안기능 요구사항에 대한 논리식을 정리형태로 증명할 수 있도록 변환한 형태이다.

```
KCOS_SPM
ACCESS_CONTROL_SP
SHIELD_ACTION_SP
INTEGRITY_PROTECTION_SP
IDENTIFICATION_AUTHENTICATION_SP
```

```
theorem Key_Block
∀KCOS_SPM • retry_counter=3 ∧ user=admin
∧auth_state=fail ⇒ system_state=key_block
```

지금까지는 정형 보안정책모델의 내부 일관성과 ST의 보안기능 요구사항들의 일관성을 확인하는 방법에 대해서 설명하였다. 이후는 SPM과 FSP의 일관성을 확인하기 위한 방법을 설명하도록 하겠다.

제일 먼저, Z로 작성된 보안정책모델을 Alloy 모델로 변환한 후에, 그림 5에 표현된 방법으로 TSF 인터페이스 모델을 Alloy 모델로 변환하는 작업을 수행하였다.



(그림 5) TSFI 모델을 ALLOY 모델로 변환 규칙

그림 5의 좌측은 일반적으로 TSFI 인터페이스가 명세, 우측은 ALLOY 명세 형식을 보여주고 있다. TSFI는 해당 인터페이스가 사용 목적, 방법과 더불어 인터페이스의 효과와 예외사항 등을 기술하고 있으며, ALLOY 명세는 모델의 동작에 대해서 동작이 발생되는 근거(FACT)와 제약조건(ASSERT) 등으로 이루어지고 있다. 우리는 인터페이스 명세의 운영모드와 TSF등에 대한 정보를 FACT로 변환하고, 효과와 예외사항 등을 고려하여 ASSERT를 명세하였으며, 해당 제약조건이 문제없이 동작하는지를 확인하기 위해서 "check assert" 명령을 통해서 확인하였다. 그림 6은 명세된 TSFI명세의 일부분이며, 명세의 정당성을 확인하기 위해서 ALLOY Analyzer v4.1.8를 사용하여 세워진 모델에 대한 검증 작업을 수행하였다. 검증 작업을 통해서 정형 보안정책모델과 준정형 기능명세서의 일관성과 완전성을 확인하여 개발자가 제공한 문서상에 문제점이 발생하고 있는지 여부를 판단할 수 있게 되었다.

(그림 6) TSFI ALLOY모델 (일부분)

일련의 작업을 통해서 우리는 정형 보안정책모델과 준정형 기능명세서 사이의 일관성 및 완전성을 확인할 수 있게 되었다. 그 결과 우리는 수행된 평가에 대한 신뢰도를 향상시킬 수 있게 되었다.

5. 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. 현재에도 고등급 평가는 소수의 정보보호 선진국에서만 평가가 이루어 지고 있는 실정이다. 이런 상황에서 국가 기술력 확보 및 정보보호수준 향상을 위해서 고등급 평가 기술 확보는

매우 중요한 이슈이기도 하다.

본 고의 도입부에서는 정보보호 제품을 평가 기준인 공통평가기준과 그 안에서 고 신뢰도의 제품을 개발, 평가할 수 있는 정형기법의 상호관계를 설명하고, 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 이전 기준 버전인 v2.3과 앞으로 평가의 지준 버전이 될 v3.1을 비교하며 신규 버전에서 고등급 평가방법을 개발하기 위해서 고려해야 되는 사항이 무엇인지 살펴본 후에, v3.1 고등급 평가가 기존의 버전과 어떠한 부분이 상이하며, 우리가 평가 기술력을 확보하기 위해서 어떠한 부분에 초점을 맞추어야 하는 지를 확인하였다. 이를 통해서 v3.1 고등급 평가에서 사용할 수 있는 시스템의 일관성과 완전성을 확인하는 방법을 제시하였다.

현재, 평가적체가 발생할 정도로 정보보호 산업 전반에 대한 관심과 정보보호제품 개발이 증가하고 있는 시점에서 향후 국가기간 망에 대한 보다 안전한 정보보호와 신뢰도가 높은 정보보호제품 개발을 위하여 고등급 평가와 관련된 기술에 대한 지속적인 연구와 관심이 필요하다고 판단되며, 이를 위해 정책과 제도적인 측면의 접근도 필요하다고 생각되며, 이후 상위 등급과 다양한 방법론에 대한 기술 확보가 절실히 필요하다는 말로 마무리 하겠다.

참 고 문 헌

[1] [cc06] “Common Criteria for Information Technology Security Evaluation”, Ver 3.1, September 2006, CCMB-2006-09-003

[2] [cc05] “Common Criteria for Information Technology Security Evaluation”, Ver 2.3, August 2005, CCMB-2005-08-003

[3] [fra98] 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

[4] [mar05] Mark S. Merkow and Jim Breihaupt, Computer security assurance using the common criteria, Thomson/Delmar Learning, Clifton Park, NY, 2005

[5] [dav96] David L, Dill. John Rushbv. “Acceptance of Formal Methods : Lessons from Hardware Design.”, IEEE Computer. April 1996. Vol.29, No.4, pp.16-30

[6] [Jim96] Jim Woodcock, Jim Davis., “Using Z : Specification, refinement, and proof.”Prentice Hall Europe 1996

[7] [Boo99] G. Booch, J. Rumbaugh, I. Jacobson., The Unified Modeling Language User Guide, Addison-Wesley, 1999.

[8] [Saa96] M. Saaltink., The Z/EVES 2.0 User’s Guide, TR-99-5493-06a, ORA Canada, 1996.

[9] [Dan06] Daniel Jackson, “Software Abstraction : Logic, Language, and Analysis”, The MIT Press, 2006

◐ 저 자 소 개 ◑

노 병 규

1988년 충남대학교 전산학과 학사
1995년 충남대학교 대학원 전산학과 석사
2006년 순천향대학교 대학원 전산학과 박사
1988～1997 한국전자통신연구원
1997～현재 한국정보보호진흥원 보안성평가단 단장

유 희 준

1997년 고려대학교 컴퓨터학과 학사
1999년 고려대학교 대학원 컴퓨터학과 석사
2005년 고려대학교 대학원 컴퓨터학과 박사
2005 ～2007 삼성전자 정보통신총괄 무선사업부 책임연구원
2007 ～현재 한국정보보호진흥원 보안성평가단 선임연구원