Journal of Knowledge Information Technology and Systems(JKITS), Vol. 15, No. 6, pp. 943-953, December 2020
https://doi.org/10.34163/jkits.2020.15.6.003

Journal of Knowledge Information Technology and Systems

ISSN 1975-7700 (Print), ISSN 2734-0570 (Online)

http://www.kkits.or.kr

An Implementation System Design and Correctness Checking of DSDC-MAC Model

Chee-Yang Song[1], **Soon-Bok Lee**[*2]

[1]*Department of Software, Kyungpook National University, Korea*
[2]*Headquaters, ROK Navy, Korea*

ABSTRACT

In earlier study, the DSDC (Duties Separation & Data Coloring) -MAC model that applied the separation of duty (SoD) and data-coloring security techniques as a unified policy was provided. For practical use of this model, the implementation system must be designed and the model corrected. To do this, this paper proposes to design an architecture for an implementation system based on the DSDC-MAC model and check the accuracy of this model. First of all, the design for the implementation system identifies the necessary functional modules based on the DSDC-MAC model and structure them into design the architecture. Subsequently, the operating process of how the module-to-module access interactions are made during execution was designed. As a case study, the design of the actual implementation system is shown by applying the proposed architectural model for the MAC model of the HRMS (Human Resources Management System). Next, to verify the accuracy of the model, the model is formalized using the Z language and examined through the Z/EVES tool. This enables the system implementation of the data security access model using the DSDC-MAC model. Furthermore the accuracy of the DSDC-MAC model structure was checked to confirm that the model is correct.

KEYWORDS: DSDC-MAC model, Implementation system design, Model checking, Z formal specification, Data coloring access control.

ARTICLE INFO: Received 2 November 2020, Revised 18 November 2020, Accepted 11 December 2020.

[*]Corresponding author is with the Department of Software, Kyungpook National University, 2559, Kyeongsang Dae-ro, Sangju-Si, Gyeongsang Buk-Do, 37224, KOREA.
E-mail address: cysong@knu.ac.kr

1. 서 론

IT(Information Technology) 기술의 지속적 성장 가속화, 정보와 솔루션의 다양화 그리고 산업간 융합화에 기인하여 안전한 서비스의 제공이 강하게 요구되는 실정이다. 컴퓨팅(computing) 환경이 웹 기반 소프트웨어에서, 스마트폰 성장과 함께 모바일 앱[1]으로, 클라우드 컴퓨팅 기반의 서비스 형태[2]로 진화하고 있다. 특히, 클라우드 서비스는 보안성의 유지[3-4] 및 사이버 안전[5-7]이 중요하다. [8]에서, 사용자의 민감한 정보 및 개인 정보의 누수를 방지하기 위해 privacy(개인 혹은 회사 정보) 기반의 클라우드 서비스 개발을 위한 가이드를 제시했다. 아울러, [9-11]에서 클라우드 서버에 저장된 데이터의 삭제 보증을 위해 데이터 삭제 보안 모델을 정립하고, 클라우드 데이터의 보증된 삭제를 위한 속성 기반의 새로운 암호화 기법을 제안하기도 하였다. 또한, [12]에서 악의적 내부자에 의한 공격을 사이버 위협의 중요한 이슈로 언급했다. 발전하는 내부자의 비정상 행위의 위협을 식별하기 위한 좀더 강력한 접근제어의 보안기술이 필요하였다.

이에, [13]에서 데이터에 대한 기밀성(Confidentiality, Privacy)과 무결성(Integrity)을 동시에 보장하기 위한 MAC(Mandatory Access Control) 모델을 확장한 DSDC(Duties Separation & Data Coloring)-MAC 모델을 제시했다. 직무분리와 데이터 컬러링(Data-coloring) 기법을 이용해서 BLP(Bell and LaPadula)와 Biba의 MAC 모델에 기밀성과 무결성을 동시에 보장할 수 있는 보안 정책과 모델을 제시했다. 그러나, 보안 모델만 제시하여 실질적인 사용을 위해서는 모델의 정확성에 대한 검증이 필요하고, 이 모델에 기반해서 보안 시스템을 설계하고 구현을 통해서 지원되어야 한다. 한편, DSDC-MAC 모델은 비 정형적인 시각화 모델인 클래스 모델로 표현되었다. 이에, 구조물을 구성하는 요소들간에 구문적 및 의미적으로 불일치하는 충돌은 없는지 검사가 필요하며, 각 요소의 불변적 사항들이 명확하게 명세가 되어야 한다.

본 연구는 [13](저자에 의해 작성)의 확장 논문으로, DSDC-MAC 모델에 기반한 구현 시스템의 아키텍처를 설계하고, Z 언어를 사용해서 모델을 정형적으로 명세하고 Z/EVES 도구를 통해 정확성을 검사한다. 먼저, 구현 시스템의 설계는 DSDC-MAC model에 기반해서 필요한 기능 모듈을 식별해서 이들을 구조화하여 아키텍처를 디자인한다. 이어서, 이 시스템 아키텍처의 모듈들이 어떻게 상호작용하여 데이터 접근의 보안 인증 과정이 수행되는 것인 지에 대한 실행시의 모습을 정립한다. 적용사례로서 인사관리시스템(HRMS: Human Resources Management System)[13]의 MAC 모델을 가지고 제시된 아키텍처 모델을 적용해서, 실제 구현 시스템의 설계를 보인다. 다음으로 DSDC-MAC 모델의 정확성을 입증하기 위해, Z[14]를 사용하여 정형적으로 명세[15-16]하고, Z-Eves 검증 도구[17]를 사용해서 명세된 Z 스키마를 검사(checking)를 수행하여 모델의 정확함을 확인한다[16, 18-19].

본 논문은 다음과 같이 구성된다. 2장에서는 관련 연구로 기 제시된 DSDC-MAC 모델과 Z 언어에 대해 살펴본다. 3장에서는 DSDC-MAC 모델을 시스템으로 구현하기 위한 아키텍처의 설계를 기술한다. 4장에서는 DSDC-MAC 모델에 대해 Z로 명세하고 Z/EVES로 모델 체킹한다. 마지막으로 5장에서 결론을 맺는다.

2. 관련 연구

2.1 DSDC-MAC 모델

DSDC-MAC 모델[13]은 직무분리와 데이터 컬러

링을 이용해서, 주체(Subject)인 user에게 그리고 객체(object)인 data에게 보다 세분화된 접근제어를 제공하여 기밀성과 무결성을 동시에 보장하는 정책을 지원한다. 보안 정책에 따라 주체(User)와 객체(Object)에 대해 직무를 분리하고, 이렇게 분류된 주체별로 data의 보안 등급에 따른 color를 부여하였다. 부여된 color별로 보안키를 매칭(matching)하여 주체가 객체(혹은 데이터)를 접근 또는 접근하지 못하도록 제어한다. 이를 통해, MAC 모델 기반 기밀성과 무결성의 동시 지원시 보안 정책간 충돌을 방지할 수 있다. <그림 1>에서 처럼, DSDC-MAC 모델의 구조는 보안 등급 정책(SLP), 직무분리 정책, Data-Coloring 규칙이 적용된 주체(S), 보안키(SK), 객체(O)와의 연관 관계를 중심으로 구성되었다. <그림 1>은 [13]의 DSDC-MAC 모델에 대해서, 속성, 연산과 스테레오타입을 제거하고 관계명과 다중성을 새로이 추가하여 작성한 것이다. 그 이유는 Z 명세의 복잡성을 줄이고 스테레오 타입을 Z로 표현할 수 없기 때문이다.

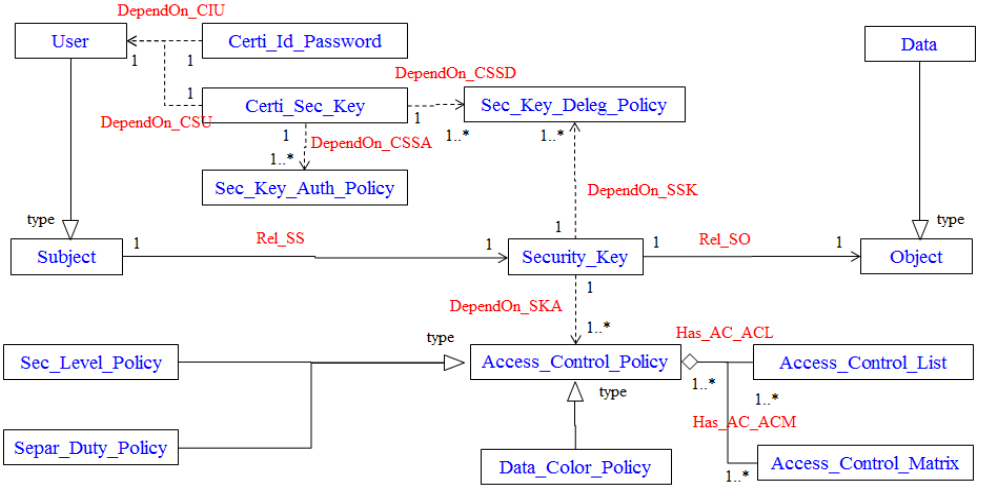

그림 1. Z 명세 위해 수정된 DSDC-MAC 모델[13]
Figure 1. Revised DSDC-MAC model[13] for specifying in Z

2.2 Z 언어와 Z/EVES 검증 도구

모델, 메타모델 혹은 프레임워크가 가진 구문과 의미를 정형적으로 명세하기 위한 언어 중에 Z 언어[14]가 있다. Z는 범용적 ISO 표준 언어로서 증명(proof)을 위한 좋은 기능을 제공하며, Z의 state 스키마가 메타모델을 표현한 클래스 모델의 class와 동일한 구조를 갖으며, 정적 속성(property)을 갖고 있어 상호 직접적 변환을 할 수 있다. Z로 명세된 Z 스키마의 구문적 및 의미적 검사를 도구로 Z/EVES Tool[17]이 널리 이용되고 있다. [16, 18-19]에서 제시된 프레임워크 혹은 메타모델을 대상으로 Z 언어로 변환, 명세하고, 이 Z 스키마 명세를 Z/EVES 도구를 사용해서 검사를 보였다.

3. DSDC-MAC 모델의 구현

본 장에서는 기존 DSDC-MAC 모델에 기반한 시스템의 구현을 위한 아키텍처를 설계하고, 이 아키텍처 모델에 따라 접근 제어에 관한 보안 인증의 실행 절차 그리고 적용사례를 다룬다.

3.1 DSDC-MAC 구현 모델의 아키텍처

제시한 DSDC-MAC 모델을 기반해서 구현 시스템의 아키텍처를 설계한 것이 <그림 2>이다.

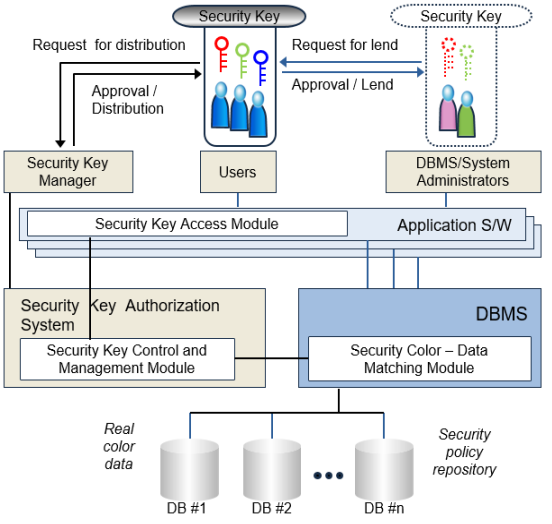

그림 2. DSDC-MAC 모델 기반 구현 시스템의 아키텍처 설계
Figure 2. Architecture design of implementation system based on DSDC-MAC model

접근제어의 동작과정에 대해서, 주체인 사용자와 data인 객체간을 연결하는 보안키는 사용자가 보안키 관리자로부터 승인받아 배부 받는다. 데이터베이스 관리시스템(DBMS: Data Base Management System) 혹은 시스템 관리자는 필요시 해당 객체의 주체인 사용자에게 대여요청 후, 승인받아 데이터에 대한 접근을 수행한다. 주요 핵심모듈로 보안키 접속 모듈, 보안키 인증 시스템내에 존재하는 보안키 통제 관리 모듈, DBMS 내에서 처리하는 보안컬러-데이터 매칭 모듈이 있다. DBMS 내에는 데이터 컬러링된 객체들이 보관되어 사용자가 해당되는 접근 권한을 가진 보안키로 접근시에 인증 절차를 거쳐 읽기, 쓰기 등의 접근이 이루어진다. 이러한 보안 구조를 가지는 DBMS와 연계되는 응용 소프트웨어(application software)들은 연동을 위한 보안키 접속 모듈을 추가하여야 한다. 이에 따른 응용 소프트웨어의 변경 비용이 발생할 것이나, 보안키 인증 시스템에서 간단한 에이전트 형식으로 개발을 한다면 그 비용은 최소화시킬 수 있을 것이다.

DSDC-MAC 모델 구현 시스템에서 주요 모듈의 세부적 처리 기능은 다음과 같다.

Security_Key Access Module은 응용 소프트웨어 개발시에 포함해야 하는 모듈로, 사용자가 가진 보안키에 대한 인증 값을 보안키 통제 관리 모듈에 전송하여 인증을 받아서, 사용자가 해당 보안컬러가 적용된 데이터에 접근하는 권한을 수행할 수 있도록 하는 기능을 제공한다. 사용자 ID & Password 인증 방식과 연계하여 2차 인증 수단(이중 시건 장치)으로서 보안키가 이용되도록 적용되어야 한다.

Security_Key Control Management Module은 사용자가 가지는 보안키에 대한 통제 및 관리 기능을 제공한다. 이것은 DBMS 내에 보안키 관리 모듈로 혹은 외부에 별도 모듈로 구현할 수 있다. 이 모듈은 응용 소프트웨어 내의 보안키 접속 모듈로부터 입력된 사용자의 보안키 입력 값을 인증하여 정당한 권한 수행을 할 수 있도록 통제하는 기능과 보안키 위임 신청 및 승인과 관련된 관리 기능을 수행한다. 또한, Security Color-Data Matching Module과 연계하여 DB에 접근할 수 있다.

Security Color-Data Matching Module은 DBMS에 포함되어야 하는 핵심이 되는 기능으로, 데이터 컬러링 Rules에 따라 Database의 각 Field 등에 대한 보안컬러를 정의하고, 각각의 보안컬러와 보안키를 매칭하는 기능을 수행한다. 또한 별도의 정책 저장소(Policy Repository)에 저장된 ACL, ACM 그리고 ACP와 SLP, SoD P에 따라 세부적인 접근 제어 기능을 수행한다. 즉, 인증된 사용자의 보안키별로 권한(조회, 쓰기, 수정, 삭제)을 수행하도록 제어한다. JOIN 또는 VIEW 수행 시에도 데이터와 보안컬러에 대한 일관성을 유지한다.

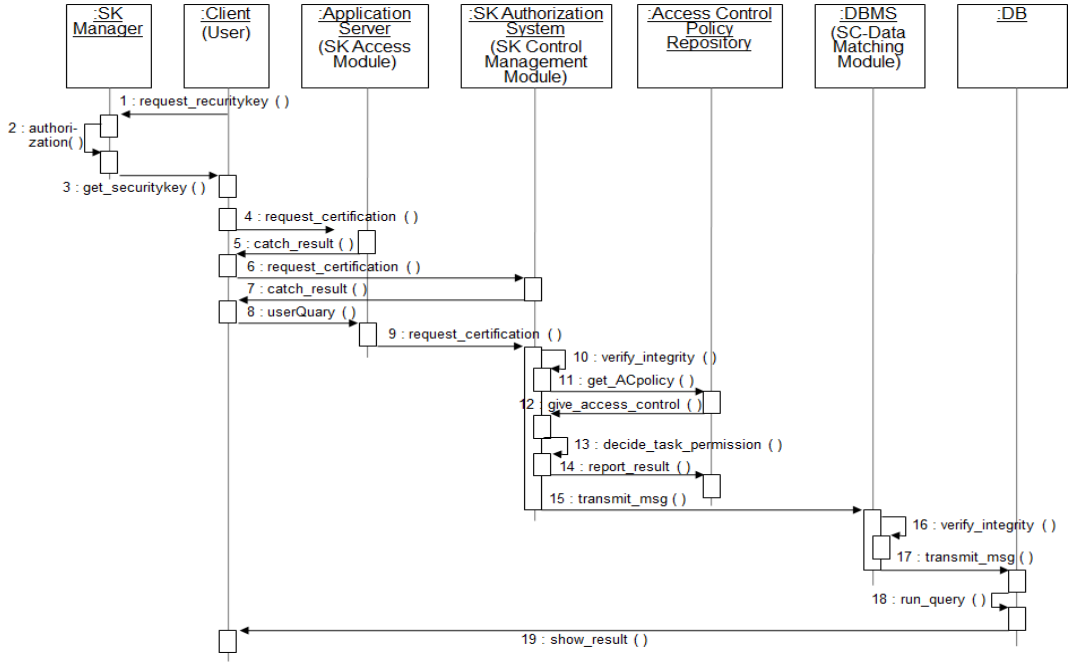

그림 3. DSDC-MAC 구현 시스템 아키텍처 기반의 접근 제어 과정
Figure. 3 Access control process based on DSDC-MAC implementation system architecture

3.2 DSDC-MAC 구현 시스템의 접근 제어 절차

<그림 2>의 DSDC-MAC 구현 시스템의 설계 모델에서, 어떻게 기능 모듈들이 상호작용해서, 직무 분리된 사용자(주체)가 보안키를 배부 받아, 데이터 컬러링된 DB(data)에 접근 제어하는 절차를 표현한 것이 <그림 3> 이다.

<그림 3>에서, 과정 1~3은 사용자가 보안키 관리자로부터 해당 레벨에 맞는 보안키를 할당받고, 보안키 접속모듈이 포함된 응용 소프트웨어 서버는 과정 4~5와 같이 ID와 PW를 기반으로 1차 인증과정을 거치고, 사용자의 보안키 값을 이용해 과정 6~7과 같이 보안키 통제모듈을 통해 2차 인증을 실시한다. 이후 사용자의 검색 질의가 있으면, 과정 8~9와 같이 응용 소프트웨어의 보안키 접속 모듈은 보안키 인증 시스템으로 해당 값을 전송하고, 보안키 인증 시스템은 과정 10~14와 같이 자체 무결성

검증 과정을 거쳐 해당 사용자의 보안컬러 레벨과 접근 제어 정책과의 비교, 분석을 통해 결정된 접근 권한을 통보받게 된다. 이 과정에 대한 결과 값은 접근제어 정책 저장소에 저장되게 되고, 추후 검토 데이터로 활용된다. 이렇게 할당된 접근권한을 바탕으로 과정 15~17과 같이 DBMS 내 보안컬러-데이터 매칭 모듈은 보안컬러와 객체와의 접근권한에 대한 무결성 검증을 거쳐 DB로 메시지를 전달하고, 과정 18~19에서 최종적으로 질의 결과를 클라이언트에게 전달하게 된다.

3.3 DSDC-MAC 모델 기반 HRMS 구현 시스템 설계

[13]의 적용 사례에서 제시된 HRMS MAC model을 가지고 단원 3.1에서 제시한 DSDC-MAC 구현시스템의 아키텍처인 <그림 2>를 적용해서 구현 시스템을 설계한 것이 <그림 4>이다.

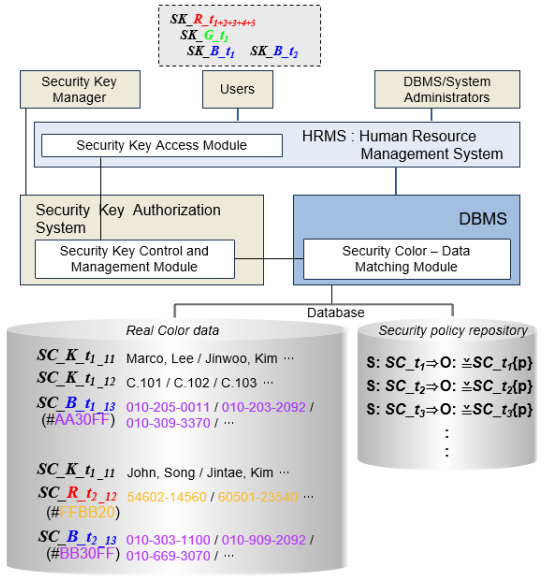

그림 4. DSDC-MAC model 기반 HRMS 구현 시스템 설계
Figure 4. HRMS implementation system design based on DSDC-MAC model

<그림 2>에서 설명한 바와 같이, 사용자, 즉 주체는 해당직무 및 보안등급에 맞는 보안키(예, $SK\_G\_t_1$, $SK\_B\_t_1$)를 보안키 관리자로부터 수령하고, HRMS에 접속하여 보안키를 인증 받은 후 보안키에 적합한 데이터에 접근한다. 이때, 데이터 접근시에는 Security Color-Data Matching module을 통해 Security policy repository의 정책 적용을 받는다.

4. DSDC-MAC 모델의 정형 명세 및 모델 체킹

클래스 모델의 비정형적으로 명세된 DSDC-MAC 모델(<그림 1>)은 이 구조물이 가진 구문(Syntax)과 의미(Semantic)를 정형적으로 명세하고 모델을 정확성을 검증하여야 한다. 본 장에서는 이 모델의 구조물을 정형적 언어인 Z[15-16]로 명세하고, 이 Z 명세를 가지고 Z-Eves[17] Tool을 이용한 모델 검사를 다룬다[18-19].

4.1 DSDC-MAC 모델의 정형 명세

본 단원에서는 Z 언어를 사용하여 DSDC-MAC 모델을 정형적으로 명세한다. 즉, 모델을 표현한 시각적 명세 표기의 클래스 다이어그램을 정형적 명세 표기로의 구문적 및 의미적 변환, 명세를 다룬다. 정형명세는 [15-16]에서 제시한 클래스 메타 모델과 Z간의 변환 규칙에 의해서 DSDC-MAC 모델의 구문과 정적 의미(Static Semantics)에 대해 Z 스키마로 명세한다.

[15-16]에 제시된 클래스 모델의 Z로의 변환 규칙을 이용해서 <그림 5>의 모델을 대상으로 Z로 정형화하여 명세한다. 이 변환 규칙은 타입 선언하고, 이 타입에 기반해서 클래스 대상의 요소 기반 Z 스키마를 정의하고 클래스들간의 관계에 기반한

Z 스키마를 명세한다[16, 18-19].

Z 명세의 구축 결과로서, 기본타입으로 subject_ID외 11개의 기본 타입을 선언하고, 요소 기반 스키마 명세로 Security_Key외 14개를 정의하고, 그리고 관계 기반 스키마 명세로 Depend_CIU 외 9개를 작성하였다.

4.1.1 타입 선언

기본 타입들의 선언은 기본 타입과 자유 타입으로 기술한다. 기본 타입의 선언은 프레임워크 메타 모델에 포함된 모든 요소들에 대해서 각 요소별로 기본 타입을 명세한다. 상속 관계를 갖는 단말 요소들에 대해서는 기본 타입을 선언하지 않고, 각 타입에 대해서 자유 타입으로 선언한다.

기본 타입의 선언은 다음과 같다.
[Subject_ID, User_ID,
Certi_Id_Password_ID,
Sec_Key_Auth_Policy_ID,
Sec_Key_Deleg_Policy_ID,
Certi_Sec_Key_ID,
Access_Control_Matrix_ID, label_ID,
Access_Control_List_ID,
Access_Control_Policy_ID,
Security_Key_ID, Object_ID]

4.1.2 Z 스키마 명세

Z의 스키마 명세는 DSDC-MAC 모델에 포함된 모든 요소에 기반해서 그리고 관계에 기반해서 각각 Z 스키마로 명세한다. 요소 기반의 Z 스키마 명세는 한 개 요소(클래스)에 대해 한 개의 Z 스키마로 명세한다. 요소들간에 Z 스키마로 정의하는 순서는 상향식 방식으로 하부 요소에서 점차 상위 요소로 각 요소에 대해서 Z 스키마로 명세해 나간다[15-16]. <그림 5>는 <그림 1>의 DSDC-MAC 모델에 대한 요소 기반의 Z 스키마 명세를 보여준다.

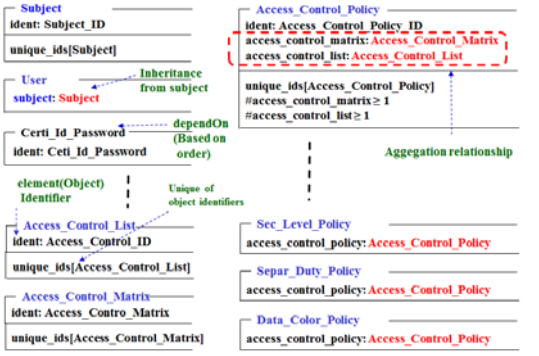

그림 5. DSDC-MAC 모델의 요소 기반 Z 스키마 명세
Figure 5. Z schema specification of DSDC-MAC model based on element

<그림 5>에서, 상속에 대해 부모 클래스인 “Subject” 스키마를 먼저 선언하고, 자식 클래스인 “User” 스키마내에 “Subject” 스키마를 변수로서 포함하여 기술한다. 다음에, “User” 클래스에 의존하는 “Certi_Id_Password” 클래스는 “User” 스키마 다음에 “Certi_Id_Password” 스키마를 명세한다. “Access_Control_Policy” 클래스는 “Access _Control_Matrix”와 “Access_Control_List” 클래스와 Aggregation의 관계를 갖는다. Z 스키마 명세는 “Access_Control_Matrix” 등의 부분 객체들을 먼저 스키마로 정의하고 “Access _Control_Policy” 전체 객체의 스키마 정의시 부분 객체들을 내부 요소로 시그네쳐부에 포함 관계로서 기술한다. 각 스키마는 스키마 명, 변수들로 구성되는 선언부와 제약조건의 Semantics를 표현하는 술어부로 구분하여 명세한다.

관계 기반의 Z 스키마 명세는 두 요소간의 관계별로 Z 스키마를 정의한다. 이 스키마 명세는 클래스간 관계 유형에 맞게 그리고 다중성(multiplicity)

을 명확하게 구분하여 정의할 수 있다. <그림 1>의 DSDC-MAC 모델에 대해서, 관계 기반 스키마 명세 10개 중에서 일부를 나타낸 것이 <그림 6>이다.

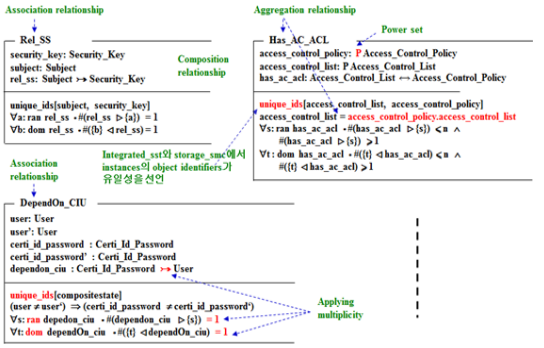

그림 6. DSDC-MAC 모델의 관계 기반 Z 스키마 명세
Figure 6. Z schema specification of DSDC-MAC model based on relationship

관계 기반 스키마의 시그네쳐부에는 각 relation 별로 연결된 요소들과 이들간의 함수관계를 선언한다. 술어부에는 다중성 (multiplicity) 및 불변 제약사항을 기술한다. 먼저, "Rel_SS" 연관 관계에 관계하는 두 개 요소는 "Subject"와 "Security_key" 이다. 연관 관계의 스키마는 시그네쳐부에 "Subject"와 "Security_key"를 선언하고, 두 번째 줄은 domain과 range에 restriction을 주어 두 요소 간의 cardinarity인 1:1을 표현한다.

"Has_AC_ACL" ' 집합 관계에서, 스키마의 시그네쳐부의 두 요소의 객체 변수 선언시, 다중성에 의해 생성할 수 있는 객체 수에 대해 단일 set(객체가 1개) 혹은 Power set(객체가 2개 이상)으로 기술한다. 술어부에서, 두번째 줄은 집합(aggregation) 관계에 대해서 "access_control_list=access_control_policy.access_control_list"로 명세를 한다. "DependOn_ciu" 종속 (dependency) 관계는 한쪽이 다른 쪽에 연향을 주는 의미적인 관계이다. 요소 "Certi_Id_Password"가 요소 "User"에 의존관계를 가지므로, 요소 "User"가 변경 이전과 변경 이후가 동일하지 않다면, 요소 "Certi_Id_Password" 도 변경 이전과 변경 이후가 같지 않도록 정의한다.

4.2 DSDC-MAC 모델의 모델 체킹

이전 단원에서 작성한 Z 명세에 대한 정확성 검사가 필요하다. 왜냐하면, DSDC-MAC 모델이 Z 스키마로 변환, 명세되었고, 이 Z 명세의 구문적 및 정적 시멘틱 속성이 정확함을 보여야 모델이 명확하다는 것을 입증하기기 때문이다. 이를 위해, Z/EVES Tool을 사용하여 Z 명세에 구문적 구조와 속성 등의 정확성 검사가 요구된다. Z/Eves Tool을 사용해서, Z 스키마 명세에 대해 syntax checking, variable range checking 및 consistency type checking을 체킹(checking)한다 [14-16].

Z/EVES 툴을 통한 검사 결과로서, 먼저, 단원 4.1.1의 기본 타입 선언 11개, 자유 타입 5개에 대한 명세에 대해 "syntax"와 "proof" 속성의 검사결과 "Y"로 실행 결과가 나왔다. 이로서 기본 타입과 자유 타입에 대한 오류가 없음을 확인하였다.

다음으로, 단원 4.1.2(<그림 5>)에 대한 요소 기반 Z 스키마 명세에 대해 15개에 대한 모델 검사의 결과를 <그림 7>에서 보여준다.

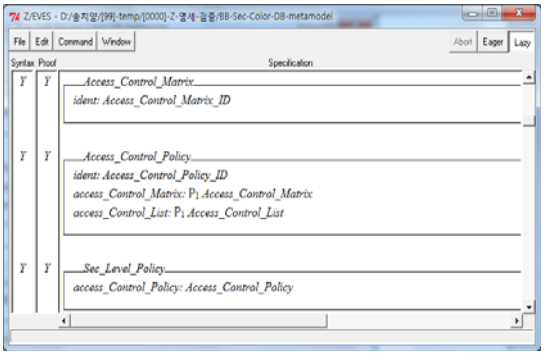

그림 7. 요소 기반 Z 스키마 명세의 구문검사 결과
Figure 7. Syntax checking result of Z specification based on element

<그림 7>에서, 왼쪽 상단에 표시된 'syntax'는 syntax와 type checking을 보여주는 것으로 'Y'로 나타나면 타입간 일치성이 있음을 입증하는 것이다. 반면, ' proof'는 domain checking의 결과를 보여주는 것으로 'Y'로 나타나면, 그 명세가 정확함을 의미한다. <그림 7>에서, 포함 관계를 갖는 "Access_Control_Policy" 스키마는 이전에 스키마로 선언된 "Access_Control_List"와 "Access_Contrl Matrix"를 포함하는 것을 술어부에 instance로 명세하였다. 상속 관계에 대해서, "Sec_Level_Policy" 스키마가 앞서 정의한 "Access_Control_ Policy" 스키마를 상속받아 정의됨을 술어부에 기술하였다.

끝으로, 단원 4.1.2(<그림 6>)에 대한 관계 기반 Z 스키마 명세인 10개(의존 6개, 연관 관계 2개, 집합 관계 2개)에 대한 모델 체킹의 결과를 <그림 8>에 나타내었다.

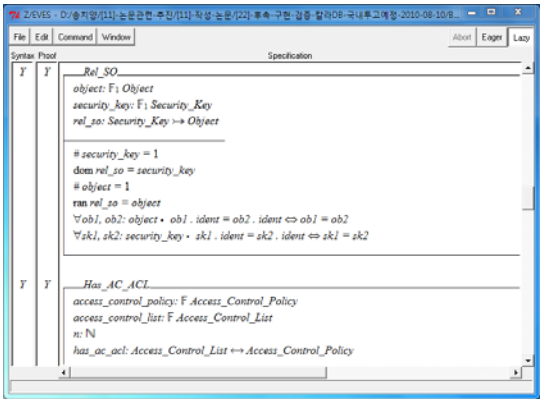

그림 8. 관계 기반 Z 스키마 명세의 구문검사 결과
Figure 8. Syntax checking result of Z specification based on relationship

<그림 8>에서, syntax와 proof 부문이 "Y" 의 결과를 보임으로서 명세가 정확함을 알 수 있다. Proof는 reduce에 의해 prove하였다(prove by reduce). 그러나, 의존 관계의 스키마 경우, <그림 5>의 " DependOn_CIU " 스키마에서 "(User≒User') → (Certi_Id_Password≒Certi_Id_Password')에 대해 Z/EVES에서 표현할 수 가 없어서 술어부에 의존을 표현하고, 술어부에 대응수와 유일성만을 표현하여 검사하였다.

결과적으로, 이와 같은 모델 체킹을 통해, DSDC-MAC 모델의 구문적 및 정적 시멘틱 측면에서 오류가 없음을 입증할 수 있다. 즉, 클래스간의 관계와 대응 수를 명확하게 명세하고, 검사하여 메타모델의 구문적 구조에서 요소들간 불일치 등의 오류(error) 혹은 충돌(conflict)이 발생하지 않았다.

5. 결 론

기존 DSDC-MAC 모델은 실용적 사용을 위하여 모델이 검증되어야 하고, 시스템으로 설계되고, 구현되어 지원되어야 한다. 본 고에서는 DSDC-MAC 모델 기반의 시스템 구현을 위한 아키텍처 모델을 설계하였다. 모델에 필요한 기능을 도출해서 독립된 모듈로 만들어 이들을 보안 정책과 인증절차에 따라 아키텍처로 설계하고, 실행시에 어떻게 모듈간 접근의 상호작용이 이루어지는지 실행 아키텍처의 동작 과정을 설계하였다. 제시한 구현 시스템의 아키텍처 모델에 대해서 신뢰성을 주기 위해 인사관리시스템(HRMS)의 MAC 모델을 대상으로 구현 시스템을 설계해 보였다. 아울러, DSDC-MAC 모델에 대해서 Z 언어를 사용해서 정형적으로 명세를 하였고, 이 Z 명세서를 Z/Eevs에 입력해서 모델을 체킹하여 정확함을 보였다. 이를 통해, DSDC-MAC 모델을 이용한 데이터 보안 접근의 시스템 개발을 가능하게 하였다. 또한, DSDC-MAC 모델의 구조물을 검사하여 모델이 정확함을 입증하였다. 향후 연구로는 제시된 DSDC-MAC의 시스템 구현을 위한 설계 모델이 어플리케이션 개발의 보안을 위한 지원도구로서 사용될 수 있도록 S/W로 구현되어야 할 것이다.

References

[1] E-S. Cho, *Design of a metamodel for the development process of a mobile application*, Korea Academy Industrial Cooperation Society, Vol. 15, No. 8, pp. 5248-5255, 2014.

[2] C-Y. Song, and E-S. Cho, *A service-oriented cloud modeling method and process*, International Journal of Electrical and Computer Engineering(IJECE), Vol. 10, No. 1, pp. 962-977, 2020.

[3] Samjong KPMG, *Analyzing domestic cloud adoption issues: Focusing on the policy of major countries*, Economic Research Institute, 2016. https://home.kpmg.com/kr/ko/home/insights/2016/05/issue-monitor-_52--.html

[4] C-Y. Song, and Y-H. Kim, *A software modeling method for integrating functional and security design*, Journal of Knowledge Information Technology and Systems, Vol. 12, No. 1, pp. 131-155, 2017.

[5] Y-S. Yun, and G-I. An, *A survey on public awareness of cyber security*, Korea Software Assessment and Valuation Society, Vol. 15, No. 1, pp. 87-95, 2019.

[6] H-H. Kim, Y-K. Kim, and K-G. Doh, *Model based vulnerability analysis for SOA*, Korea Software Assessment and Valuation Society, Vol. 15, No. 1, pp. 87-95, 2012.

[7] H-M. Jung, K-S. Han, and G-S. Lee, *A schema design for supporting the cyber security control of SCADA*, Journal of Knowledge Information Technology and Systems, Vol. 7, No. 6, pp. 174-179, 2012.

[8] S. Pearson, *Taking account of privacy when designing cloud computing services*, HP Laboratories, 2009.

[9] L. Xue, Y. Yu, Y. Li, M. H. Au, X. Du, and B. Yang, *Efficient attribute-based encryption with attribute revocation for assured data deletion*, Information Sciences, 2019.

[10] N. Elmrabit, S. H. Yang, L. Yang, and H. Zhou, *Insider threat risk prediction based on bayesian network*, Computers & Security, 2020.

[11] L. Liu, C. Chen, J. Zhang, O. De Vel, and Y. Xiang, *Insider threat identification using the simultaneous neural learning of multi-source logs*, IEEE Access, 2019.

[12] Y. Wang, L. Tian, and Z. Chen, *Game analysis of access control based on user behavior trust*, Information 2019, MDPI, 2019.

[13] S-B. Lee, Y-H. Kim, J-W. Kim, and C-Y. Song, *A design of MAC model based on the separation of duties and data coloring: DSDC-MAC*, Journal of Computer Science (JCS), Vol. 16, No. 1, pp. 72-91, 2020.

[14] J. M. Spivey, *The Z notation - a reference manual*, New York: Prentice-Hall, 1989.

[15] M. Shroff, R. France, *Towards formalization of UML class structures in Z*, in Proc. of the COMPSAC '97, Washington DC, pp. 11-15, 1997.

[16] C-Y. Song, *A metamodel-based modeling mechanism for hierarchical design in UML*, Thesis for the Degree of Doctor, 2003.

[17] M. Saaltink, *The Z/EVES 2.0 users guide*, TR-99-5493-06A, ORA Canada, 1999.

[18] C-S. Cho, C-J. Kim, and C-Y. Song, *A*

formal specification of reusable framework of embedded system, Korea Information Processing Society, Vol. 17-D, No. 6, pp. 431-442, 2010.

[19] E-S. Cho, and C. Y. Song, *A formal specification and meta-model for development of cooperative collection ·analysis framework*, Journal of The Korea Society of Computer and Information Vol. 24 No. 12, pp. 85-92, 2019.

DSDC-MAC 모델의 구현 시스템 설계 및 정확성 검사

송치양[1], 이순복[2]

[1] *경북대학교 소프트웨어학과 교수*

[2] *한국 해군본부 중령*

요 약

기존에 BLP model과 Biba model의 보안정책을 하나의 통합된 정책으로 직무분리(SoD: Seperation of Duty)와 Data-Coloring 보안 기법을 적용한 DSDC (Duties Separation & Data Coloring)-MAC 모델을 제시했다. 이 모델의 실제적인 사용을 위해서, 구현 시스템을 설계하고, 모델의 정확성이 입증되어야 한다. 본 연구에서는 DSDC-MAC 모델 기반의 구현 시스템을 위한 아키텍쳐를 설계하고, 이 모델의 정확성을 검사한다. 먼저, 구현 시스템을 위한 설계는 DSDC-MAC model에 기반해서 필요한 기능 모듈을 식별해서 이들을 구조화하여 아키텍처를 디자인한다. 이어서, 실행 시에 어떻게 모듈간 접근의 상호작용이 이루어지는지의 동작 과정을 설계하였다. 적용사례로서, 인사관리 시스템(HRMS: Human Resources Management System)의 MAC 모델을 대상으로 제시된 아키텍처 모델을 적용해서, 실제 구현 시스템의 설계를 보인다. 다음으로, 이 모델의 정확성을 입증하기 위해서, Z 언어를 사용해서 모델을 정형적으로 명세하고 Z/EVES 도구를 통해 검사한다. 이로서, DSDC-MAC 모델을 이용한 데이터 보안 접근 모델의 시스템 구현을 가능하게 하였다. 아울러, DSDC-MAC 모델 구조물의 정확성을 검사하여 안정적인 모델임을 확인하였다.

Chee-Yang Song received the bachelor and master's degrees in Computer Science from the Hannam university in 1985 and the Chungang University in 1987, respectively. He received the Ph.D. degree in Computer Science from Korea University in 2003. From 1990 to 2004, he was a researcher Research center of Korea Telecom (KT). He has been a Professor in the Department of Software at Kyungpook National University since 2008. His research interests include service oriented modeling, business-software integrated design process, component based software engineering, and security design.

E-mail address: cysong@knu.ac.kr

SoonBok Lee received the M.S. degree in Computer Science from Korea University in 2008 south Korea. He is currently a Navy Officer of ROK Navy. His research interests include Data Security, Business and Software Modeling Process, Ontology, Software Product Line Engineering.

E-mail address: lsb0510@gmail.com