역할기반접근제어를 기반으로 한 분산 하이퍼텍스트 시스템 보안모델

정 철 윤[†] · 이 형 효[††] · 노 봉 남[†††]

요 약

멀티미디어, 인터넷 환경에서 하이퍼텍스트 시스템의 활용이 일반화됨에 따라 하이퍼텍스트 시스템에 저장된 정보에 대한 보호를 위해 권한부여나 접근제어와 같은 상위수준의 보안 메카니즘의 필요성이 요구되었다. 또한 분산환경에서는 하이퍼텍스트 시스템내에 저장된 정보들의 정형화된 스키마의 부재, 비체계성 등으로 인하여 보다 복잡한 체계의 보안이 필요하다. 본 논문에서는 분산 하이퍼텍스트 시스템 보안모델을 설계하기 위해 하이퍼텍스트 시스템의 특성 및 현재까지 제안된 보안 메카니즘을 살펴보고, 하이퍼텍스트 시스템상에 다양한 접근제어 정책들을 적용하였을 때의 문제점들을 제시한다. 또한 본 논문에서 제안하는 분산 하이퍼텍스트 시스템 보안모델의 기본개념인 연산도메인의 구성요소 및 관련 성질에 대해 기술하고, 현재 상용시스템에서 보안 메카니즘으로 널리 활용중인 역할기반 접근제어 정책과 연산도메인이 결합된 새로운 보안모델에 대해 기술한다. 마지막으로 본 모델의 장 단점 및 추후 연구과제를 제시한다.

A Role-Based Access Control Security Model for a Distributed Hypertext System

Cheolyun Jeong[†] · Hyunghyo Lee[††] · Bongnam Noh[†††]

ABSTRACT

Almost all of the applications in World Wide Web and multimedia environments are based on hypertext paradigm. The increasing, widespread use of such paradigm requires a security-enforcement mechanisms such as authentication and access control to guarantee the secure accesses to information. The characteristics of hypertext systems such as absence of schema, unstructuredness of information etc., need a more complex security requirements for the distributed hypertext systems. In this paper, we review the characterics and security mechanisms of hypertext system and then describe problems in applying access control mechanisms to that system. Also, we introduce an operation-domain concept for a security model, and propose a new access control security model for the distributed hyptertext systems associated with role-based access control mechanism which is widely adopted in commercial environments.

1. 서 론

하이퍼텍스트 시스템(HTS : HyperText System;이하 HTS라 칭함)은 사용자들에게 정보를 구성하고 표현하기 위한 효과적인 접근 수단으로서 노드(node)와 링크(link)로 구성된 정보구조이다. 노드는 하이퍼 문서 내의 실제적인 정보의 단위로 텍스트를 저장하며 링

† 정 회 원:광주여자대학교 전산학과
†† 정 회 원:전남대학교 전산학과
††† 종신회원:전남대학교 전산학과
논문접수:1997년 12월 3일, 심사완료:1998년 2월 17일

크에 의해 다른 노드와 연결된다[9]. 하이퍼텍스트로 구성된 정보의 검색순서와 특성은 사용자의 정보이용 목적과 등급에 따라 다양하다. HTS의 개념을 확장하여 문서의 저장장소를 지역 시스템에 제한하지 않고 근거리망 또는 공중망에 연결된 원격 시스템들로 확장할 수 있다. 따라서 HTS의 개념이 정보의 저장위치와 검색과정이 무관한 정보저장개념으로 확장된다. 또한 멀티미디어 정보 사용의 확산과 인터넷 환경의 보편화로 하이퍼텍스트/하이퍼미디어 시스템의 사용이 급격히 증가되는 추세에 따라 하이퍼텍스트를 기반으로 하는 각종 시스템들은 정보의 구성, 검색, 표현등에 관한 연구 및 개발은 활발한 반면, 보안정책 관련 분야에 대한 연구는 비교적 미진한 편이다. 특히 사용자 인증(authentication)이나 데이터의 암호화(data encryption)등 하위수준 관련 보안연구 대비, 권한부여(authorization)나 접근제어(access control)와 같은 상위수준의 보안 문제들에 대한 연구는 미비한 실정이다. 보안관점에서 볼 때 HTS의 구조적 특성인 비체계성, 정형화된 스키마가 존재하지 않는 점, 그리고 노드내의 정보의 다양성과 분산환경의 특징으로 인하여 훨씬 복잡한 보안체계가 필요하게 되었다[2]. 본 논문에서는 분산 하이퍼텍스트 시스템(DHTS:Distributed HyperText System; 이하 DHTS라 칭함)상에 적합한 보안모델을 설계하고자 한다.

2. 관련연구

2.1 하이퍼텍스트 시스템

각 시스템은 고유의 특성을 지니고 있기 때문에 표준화된 모델은 없으나 Campbell & Goodman에 의해 정의된 HTS 구조를 3단계로 제안한 모델과 HTS 구조에 대한 참조모델로 사용되는 Dexter 모델이 있다. Dexter 모델은 여러종류의 HTS에서 공통적으로 사용되는 용어들에 대한 표준화된 정의와 주요 추상화된 개념들에 대해 정형적으로 기술하므로서 시스템 특성과 기능들의 표준준용 여부를 판단하는 참조모델로서 이용되며, 또한 HTS들간의 연동과 정보교환의 표준을 정의하는 기본 모델로서 활용[5]되고 있으며 다음과 같이 구성된다. 정의된 접근방식들을 통해 하이퍼텍스트 형태로 저장된 정보 또는 문서를 사용자에게 제공하여 주는 정보표현(run-time) 계층, 지역 시스템 또는 원격시스템으로의 저장위치 투명성을 제공하는 정보저장(storage) 계층, 하이퍼텍스트의 구성규칙들로 이루어진 정보구조(within component) 계층의 3계층으로 구성되어 있다[2, 5].

2.2 하이퍼텍스트 보안 관련연구

E.B. Fernandez가 제안한 보안모델은 권한부여 관점에서 분류될 수 있는 다단계 모델(multilevel model)과 접근행렬기반 모델(access-matrix based model)을 접근권한의 생성종류에 중점을 둔 보안모델인 강제적 접근제어 모델(mandatory access control model)과 자율적 접근제어 모델(discretionary access control model)을 각각 조합하므로서 다음과 같은 4가지 보안모델을 제시[4]했으며 HTS의 보안모델에 적절히 활용될 수 있다.

□다단계 강제적 접근제어 모델:접근대상 정보는 비밀등급(sensitivity level)을, 정보사용자는 인가 등급(clearance level)을 시스템 보안관리자로 부터 부여받으며, 이 등급값들의 비교결과에 따라 접근권한부여 여부가 결정된다. 주로 국방부문의 보안영역에서 활용된다[11].

□다단계 자율적 접근제어 모델:사용자는 지정된 비밀등급을 가진 정보 또는 문서를 생성할 수 있고 타 사용자들로의 접근대상 정보의 접근권한을 부여할 수 있으나 사용자에는 인가등급이 정의되어 있지 않다.

□접근행렬기반 강제적 접근제어 모델:이 모델에는 보안관리자기반 모델(administrator-based model)[12, 13], 역할기반(role-based) 접근제어의 변경모델[14], 권한부여 규칙과 사용자 그룹을 보안관리자가 지정하는 선험적(heuristic)인 보안모델[8] 등이 해당된다.

□접근행렬기반 자율적 접근제어 모델:이 모델은 상용 데이터베이스 시스템의 SQL에 적용되어 사용중인 가장 일반적인 모델이다.

P. Samarati는 DHTS상의 보안을 위해 하이퍼텍스트 문서와 그들 관계에 대한 논리적 관점을 기술하는 하이퍼레벨(hyper level)과 문서들을 정의하는 정보가 저장되는 곳을 나타내는 베이스레벨(base level)로 분류한 모델로서 노드나 객체를 생성하는 사용자가 소

유자가 되며 공유를 위해 소유자가 권한허용 여부를 결정한다[9].

복잡한 문서의 집합이 하이퍼텍스트상에 표현되기 위해서는 복잡하게 연결된 네트워크가 필요로 하게 되고, 다양한 형태의 링크가 필요함에 따라 새로운 자료구조 및 표현도구들이 요구되어진다. S.J. DeRose는 다양하고 많은 링크들을 특성에 따라 분류하므로서 시스템이 제공하는 기능들을 쉽게 파악할 수 있게 하고, 설계자가 사용자 인터페이스나 자료구조를 정의하는데 용이성을 제공하고 있다. 또한 M.S. Olivier는 V.D. Parunak이 '91년에 분류한 링크타입들에 보안을 위한 라벨(label) 및 보안등급을 부여한 강제적 접근제어의 특성을 지닌 자율적 보안 모델을 제시[8]했으며, 그외에 KMS[1], HAM[3]등이 있다.

3. 접근제어정책과 하이퍼텍스트 시스템

이 장에서는 하이퍼텍스트 시스템에 적용할 수 있는 보안모델들과 그 특징에 대해 기술한다. 강제적 접근제어 보안모델은 HTS 보안관리자에 의해 부여된 보안대상 정보인 노드 또는 노드간을 연결하는 링크에 대한 비밀등급과 링크를 통해 연결된 노드내의 정보를 접근하고자 하는 사용자에게 부여된 인가등급을 비교하여 접근권한부여 여부를 결정하는 보안모델이다. 이 모델은 정보의 접근에 대한 통제가 너무 엄격하고 보안등급을 부여대상인 노드와 링크의 표준화된 분류의 어려움으로 인하여 대부분의 연구는 특정영역상에 진행되고 있다.

자율적 접근제어는 접근을 요청한 사용자의 신원(Identification)에 근거하여 사용자가 정보에 대한 접근 권한을 자율적으로 부여하거나 철회할 수 있다는 것을 의미한다. 이것은 접근 권한의 통제가 정보의 각 소유자에 의하여 분산화되어 수행됨을 의미하지만, 이러한 통제는 보안 관리자에 의하여 중앙 집중적으로 수행될 수도 있다. 자율적 접근 제어는 DHTS상에 적합할 것으로 여겨지고 있으며 많은 연구가 진행되고 있지만 노드의 수와 사용자의 수가 많아질 경우, 접근제어 정보를 저장 및 갱신하기 위한 부담과 각 노드의 접근때마다 접근권한 부여 여부를 판단 해야하는 오버헤드가 필연적이다.

역할기반 접근 제어 정책에서는 여러 명의 사용자를 역할이라고 하는 클래스들로 그룹화 하여 그들의 임무에 근거하여 사용자들을 그룹화한 역할에 접근 특권을 부여한다. 사용자들을 클래스 단위로 취급하여 개개의 사용자에 대한 권한 부여 횟수를 줄이는 것이다[10, 15]. 따라서 본 논문에서는 위의 2가지 모델의 특성을 혼합한 모델인 역할기반 접근제어 보안모델을 적용한다.

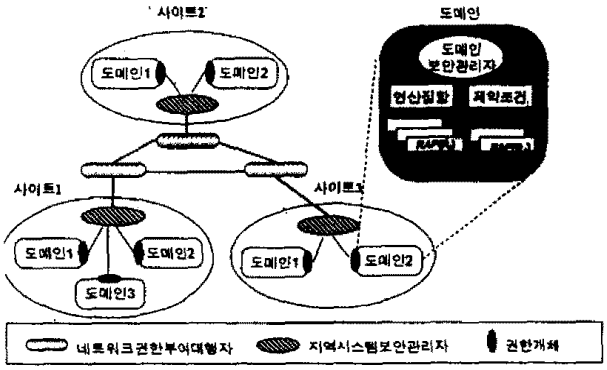
4. 분산 하이퍼텍스트 시스템 보안모델

이 장에서는 DHTS의 보안모델의 설계시 필요한 개념과 특성들에 대해 기술하고 연산도메인을 기반으로 한 분산 하이퍼텍스트 보안모델을 제안한다.

4.1 분산 하이퍼텍스트 시스템

DHTS은 정보의 저장위치의 분산환경 지원 외에 각 시스템에 적용된 보안정책, 원격시스템 사용자들로 부터의 접근요청에 대한 처리기능 등이 제공되어야 하며 DHTS 보안모델의 구성에 필요한 요소 및 적용되는 보안모델은 다음과 같다.

첫째, DHTS 보안모델의 구성요소들로서, 먼저 각 시스템에 저장된 하이퍼텍스트 정보와 원격 시스템으로 부터의 접근제어 요청 관리정보 등을 전체적으로 관리하는 지역시스템 보안관리자가 있으며, 관련된 정보와 링크의 집합인 도메인을 관리하는 도메인 보안관리자가 있다. 도메인은 본 논문에서 제안한 DHTS의 보안관리단위이며, 도메인의 구분기준은 보안대상 정보의 논리적 연관성과 정보들에 적용되는



(그림 4.1) 분산 하이퍼텍스트 시스템 보안모델
(Fig. 4.1) Security model of distributed hypertext sytem

연산종류에 의해 결정된다. 네트워크 권한부여 대행자는 원격 사용사의 접근허용 여부를 결정하는 기능을 수행한다.

둘째, DHTS 보안모델에 역할기반접근제어 모델이 사용되어지며 본 논문에서 제시하는 DHTS 보안모델의 전체 구성은 (그림 4.1)과 같다.

4.2 역할접근플레인

역할접근플레인(RAP:Role Access Plane)은 하이퍼텍스트 지역시스템보안관리자에 의해 정의된 역할 R_i가 어떤 노드에 대한 노드연산을 실행하고자 할 때 접근권한 유무에 대한 추가의 확인 절차없이 접근가능한 노드들과 그 노드들을 연결하는 링크들의 집합으로 다음과 같이 정의한다.

4.2.1 구성요소

RAP_i: 역할접근플레인

$RAP = \langle N, IRL, ERL, ISL, R \rangle$,

N(Node): 하이퍼 문서내의 실제적인 정보의 단위

IRL(Internal Role Access Plane Link): 한 역할접근플레인내에 포함된 노드들을 연결하고 있는 링크

ERL(External Role Access Plane Link): 다른 역할접근플레인에 포함된 노드와 연결된 링크

ISL(Inter System Link): 다른 사이트내 노드들을 연결하고 있는 링크

위 구성요소들인 *N*, *IRL*, *ERL*, *ISL*은 연결된 하나의 하이퍼텍스트가 아닌 역할에 부여된 접근권한에 따라 여러 개의 분할된 하이퍼텍스트의 모임으로 구성될 수 있으며, 시스템관리자에 의해 시스템내에 정의된 역할 및 역할 계층에 따라 도메인내 같은 역할을 지닌 *RAP*들이 모여 역할접근플레인 집합을 구성하는데 다음과 같이 관련 용어를 정의한다.

■ $RAP(R_i)$: 시스템내에 정의된 역할들이 R_1, R_2, R_3, ..., R_n이고 역할간에는 역할의 계층관계에 따라 역할구조가 정의되어 있을 때, 도메인 내에는 R_1, R_2, ..., R_n인 *RAP*들이 각각 존재하는데 R_i인 *RAP*을 $RAP(R_i)$로 정의하며, $RAP_1(R_i)$, $RAP_2(R_i)$, ..., $RAP_n(R_i)$이 존재할 수 있다.

■ $\boldsymbol{R}(R_i)$: 위에서 정의한 $RAP_1(R_i)$, $RAP_2(R_i)$, ..., $RAP_n(R_i)$들이 모여 다시 하나의 역할접근플레인 집합을 구성하는데 $\boldsymbol{R}(R_i)$로 정의한다.

$$\boldsymbol{R}(R_i) = \{RAP_1(R_i), RAP_2(R_i), ..., RAP_n(R_i)\},$$
$$R_i \in R, \quad i = 0, 1, 2, ..., m$$

4.2.2 역할접근플레인간의 관계

4.2.1에서 정의한 것처럼 $\boldsymbol{R}(R_i)$는 역할이 R_i인 *RAP*들의 집합으로서 정의 될 수 있으며, $RAP(R_i)$와 $RAP(R_j)$는 각각 $\boldsymbol{R}(R_i)$와 $\boldsymbol{R}(R_j)$를 구성하는 요소가 된다. $RAP(R_i)$와 $RAP(R_j)$내에 포함된 노드들은 관련성에 따라 링크에 의해 연결된다. 이때 $RAP(R_i)$와 $RAP(R_j)$에 포함된 노드들간의 관련성 유무에 따라 세가지의 구조가 존재하며, 관련성을 구하기 위한 함수 및 구조를 다음과 같이 정의한다.

임의의 $RAP(R_i)$ 내에 속하는 노드 N을 구하는 함수($\mathcal{N}$:Node)를 다음과 같이 정의한다.

$$N_i = \mathcal{N}(RAP(R_i)), \quad N_j = \mathcal{N}(RAP(R_j))$$
$$i, j = 1, ..., n$$
$$R_i, R_j \in R, \quad i, j = 1, ..., n$$

■ 구조 1: 비결합(disjoint) 관계

비결합 관계는 $RAP(R_i)$와 $RAP(R_j)$에 각각 포함되는 어떠한 노드간에도 상호 관련성이 없는 경우로서 다음과 같이 표현될 수 있다.

$$\mathcal{N}(RAP(R_i)) \cap \mathcal{N}(RAP(R_j)) \neq \phi$$
$$\forall\ i, j \quad \text{where, } R_i, R_j \in R,\ 1 \leqq i, j \leqq n$$

■ 구조 2: 중복(overlap) 관계

중복 관계는 $RAP(R_i)$와 $RAP(R_j)$에 포함되는 노드들중에 하나 이상의 노드가 $RAP(R_i)$와 $RAP(R_j)$에 의해 공유되는 경우로서 공유되는 노드들만을 포함하는 새로운 $RAP(R_iR_j)$ (또는 $RAP(R_jR_i)$)를 생성하며 링크에 의해 동시에 연결되어지는 구조로서 다음과 같다.

$$\mathcal{N}(RAP(R_i)) \cap \mathcal{N}(RAP(R_j)) \neq \phi$$
$$\forall\ i, j \quad \text{where, } R_i, R_j \in R,\ 1 \leqq i, j \leqq n$$

■ 구조 3: 포함(containment) 관계

포함 관계는 어떤 $RAP(R_i)$ 내의 모든 노드가 $RAP(R_j)$ 노드에 포함되거나 그 반대로 포함되는 경우이다.

$N_i \subseteq N_j \vee N_i \supseteq N_j$

where, $N_i \in RAP(R_i)$, $N_j \in RAP(R_j)$,

$R_i, R_j \in R$, $i, j = 1, ..., n$

4.3 도메인

도메인은 본 논문에서 제안된 DHTS의 보안관리단위로서 보안대상정보를 구성하는 역할접근플레인 집합, 도메인상에 행해지는 연산 집합, 도메인상의 제약조건, 도메인 보안 관리자 등으로 구성되며 (그림 4.2)와 같다.

D(Domain): 도메인

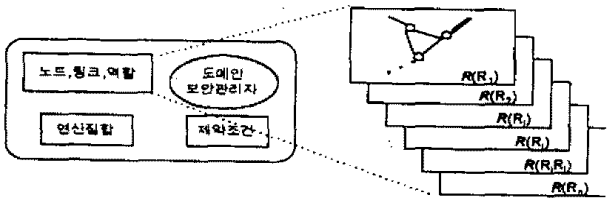
$D = \langle R, O, C, DSA \rangle$,

R(Role Access Plane Set): RAP들의 집합

O(Operation): 도메인에 실행가능한 연산

C(Constraint): 도메인 활성화를 위한 제약사항

DSA(Domain Security Administrator): 도메인 내의 정보 생성 및 변경권한을 가진 사용자



(그림 4.2) 도메인 구조

(Fig. 4.2) Architecture of domain

4.3.1 도메인 분류기준

도메인은 하이퍼텍스트 보안의 기본단위로서 그 분류기준은 하이퍼텍스트를 이용하여 정보를 저장, 표현하려는 사용자의 목적이나 정보의 논리적 분류에 의해 결정된다. 그러나 본 논문에서는 정보의 체계적인 구성기능이 구조화되어 있지 않고 각 노드에 대해 실행되는 연산이 비교적 단순한 하이퍼텍스트 시스템 환경에서 도메인의 분류기준을 관련된 노드의 집합에 적용되는 연산을 기준으로 하였다.

도메인의 분류를 노드와 링크에 실행가능한 연산을 기준으로 하면 정보의 검색이나 저작등 정보에 대한 연산의 종류가 많지 않은 HTS에서 노드들을 작은 수의 그룹으로 그룹화 할 수 있는 장점이 있다. 이렇게 분류된 도메인내의 노드 및 링크들은 $R(R_1)$, $R(R_2)$, $R(R_3)$, ..., $R(R_n)$으로 세분화 되어지며 각각에 역할기반 접근제어등을 이용한 접근제어 모델을 적용함으로써 대량의 정보를 포함할 수 있는 HTS에서의 접근제어를 효율적으로 수행할 수 있다.

4.3.2 연산의 종류 및 도메인 타입

HTS에서 사용되는 연산의 종류는 노드내에 저장된 정보의 검색, 수정을 위한 노드연산과 노드 및 링크의 생성, 삭제에 관련된 도메인연산으로 분류될 수 있다. 노드연산은 HTS의 특성 및 사용자에 제공되는 노드연산의 종류에 따라 다양할 수 있으나, 본 논문에서는 모든 하이퍼텍스트에 필수적인 검색과 수정의 두 노드연산을 기본으로 하여 기술한다.

■ 노드연산: 정보검색(view), 정보수정(modify)

■ 도메인연산: 노드 및 링크 생성(create), 노드 및 링크 삭제(delete)

주어진 연산정보를 이용하여 해당 도메인을 구하는 함수(DOMAIN: Domain of Operation Type)는 다음과 같이 정의된다.

■ DOMAIN(o) = D

D: o 연산실행을 허용하는 도메인

임의의 노드 n이 소속된 도메인 타입을 구하는 함수(DOT_by_node: Domain Operation Type by node)는 다음과 같이 정의된다.

■ DOT_by_node(n) = $\{t\}$

n: 노드, t: 도메인 타입

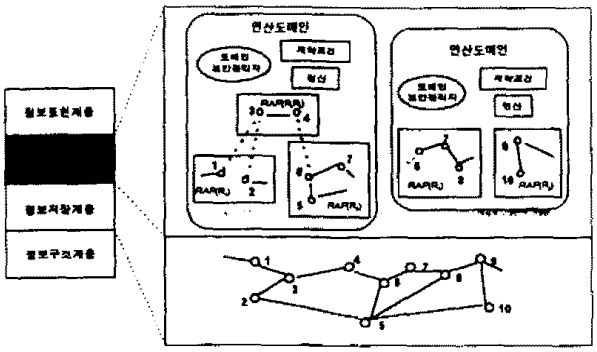
where, t v if $n \in D.N$ and $n \in D.O$

m if $n \in D.N$ and $m \in D.O$

4.4 확장된 하이퍼텍스트 시스템 구조

동일한 노드연산에 따라 분류된 도메인들로 구성된 도메인계층을 추가하고, 시스템에서 정의된 역할 계층에 따라 세분화된 $R(R_1)$, $R(R_2)$, $R(R_3)$, ..., $R(R_n)$가

도메인계층 내에 표현된 HTS의 계층구조는 (그림 4.3)과 같다.



(그림 4.3) 확장된 하이퍼텍스트 시스템 계층구조
(Fig. 4.3) Architecture of extended hypertext system

(그림 4.3)에서 볼 수 있듯이 정보저장 계층에 저장, 관리되는 노드들 중 노드에 실행 가능한 노드연산이 하나 이상인 경우, 그 노드는 하나이상의 연산 도메인에 포함된다. 그러나 어떤 노드가 한 도메인내 하나이상의 *RAP*에 포함될 경우에는 4.2.2절에서 기술된 *RAP* 간의 관계에 따라 구성됨을 알 수 있다.

4.4.1 링크 타입

노드연산을 위한 원시노드에서 목적노드로의 연결을 위해 세가지 링크타입을 정의하고 있는데, 노드 n에서 노드 n'로 연결된 링크 l의 링크타입(LT:Link Type)은 다음과 같이 정의된다.

■ LT(l) = *IRL* if, *irl*
ERL if, *erl*
ISL if, *isl*

l: 링크
IRL: *RAP* 내부 노드에 대한 링크
ERL: 다른 *RAP* 내의 노드에 대한 링크
ISL: 다른 사이트내의 노드에 대한 링크

4.4.2 역 할

HTS 보안관리자에 의해 정의 분류된 역할 및 역할계층에 따라 그룹화 되어진 *RAP*과 ***R***에 대하여 원시노드 n에서 목적노드 n'로 연결된 링크를 따라 노드연산이 행해진다. 원시노드를 포함하는 ***R***의 역할등급과 목적노드를 포함하는 ***R***의 역할등급간의 상호 역할관계에 따라 원시노드에서 목적노드로의 노드연산 가능여부가 결정된다. 따라서 어떤 노드 n 또는 노드 n'를 포함하는 *RAP*이나 ***R***의 역할을 결정해야 한다. 역할(Role)은 시스템내에 정의된 역할들과 *RAP* 간의 관계에 의해 추가로 생성된 역할 R_iR_j 등의 수만큼 존재할 수 있으며 역할을 구하는 함수는 다음과 같이 정의된다.

임의의 노드 n이 소속된 역할을 구하는 함수(Role_by_node: Role by node)는 다음과 같이 정의된다.

■ Role_by_node(n) = {t}
n: 노드, t: 역할
where, $t \in \{R_1, R_2, ..., R_i, R_j, R_iR_j, ..., R_n\}$

그리고 *RAP* 식별자에 의해 역할을 구하는 함수(Role_by_rap: Role by *RAP* 또는 ***R***)는 다음과 같이 정의된다.

■ Role_by_rap(*RAP*|***R***) = {t}
RAP: 역할접근플레인,
R: 역할접근플레인 집합,
t: 역할
where, $t \in \{R_1, R_2, ..., R_i, R_j, R_iR_j, ..., R_n\}$

4.4.3 역할 세션 문맥

사용자는 시스템 보안관리자에 의해 정의된 역할계층에 따라 부여된 역할권한을 통해 하이퍼텍스트를 순회하면서, 필요한 정보를 검색하거나 하이퍼텍스트에 저장된 정보를 수정, 추가등을 할 수 있다. 사용자에게 부여된 역할을 가지고 하이퍼텍스트 순회 시작부터 종료 까지를 역할 세션이라 하며, 역할 r의 역할 세션 문맥(RSC: Role Session Context)은 다음과 같이 정의된다.

■ RSC(r) = (s, r, n, t)
s: 사이트 식별자
r: 역할 식별자
n: 현재 순회중인 노드
t: n의 도메인타입 (t = DOT_by_node(n))

역할 세션 문맥은 노드 n에 연결된 다른 노드로의 순회나 현재 노드에 대한 노드연산 실행요청때 그 권한부여 판단의 기본정보를 제공한다.

4.4.4 노드 식별자

Dexter 하이퍼텍스트 모델에서 노드에 부여된 유일식별자를 이용하여 저장위치에 대한 정보를 계산한다. 그러나 본 논문에서 제안한 DHTS의 보안모델에서는 분산환경에서 각 노드를 유일하게 식별하기 위해 사이트 식별자와 사이트내 노드 식별자를 결합하여 사용한다.

■노드 식별자=(사이트 식별자, 사이트내 노드 식별자)

즉, DHTS에서의 노드 식별자내에는 사이트를 구별할 수 있는 유일한 식별자가 포함되어 있다.

4.4.5 도메인 보안관리자

도메인 보안관리자는 도메인을 구성하는 정보를 생성, 변경하는 기능의 수행권한을 가지며, 도메인을 구성하고 있는 노드와 링크들에 대한 보안책임을 가진다. 이때 도메인 보안관리자인 *DSA*는 시스템 보안관리자에 의해 정의된 역할 권한을 지녀야 한다. 이때 도메인 구성요소중 *O*, *DSA*, *C*에 대한 정보의 설정 및 변경은 시스템 보안관리자에 의해 이루어진다. 한 시스템내에서 도메인 보안관리자에 의한 노드 및 링크의 생성, 삭제 알고리즘은 다음과 같다.

■ 노드생성 알고리즘

```
Algorithm create_node(s, r', o, info)
begin
  // info : 생성될 노드에 저장될 정보
  // o : 생성될 노드에 적용가능한 연산
  // rsc = (s, r, n, t)
  D = DOMAIN(o)
    // 생성될 노드의 연산도메인을 구함
  if (r∈D.DSA)
    // 도메인 보안관리자

   if (r'∈D.R.R)
    // 노드가 추가될 R(r') 존재여부 확인
    // 정보구조표현층내의 노드생성함수 호출,
       생성된 노드 식별자(n') 리턴
    // 현재 도메인내의 RAP(r')에 접근제어
       정보(n') 추가
   else
    // 새로운 R(r') 생성, 노드생성, 정보추가
   fi
   for all rap ∈ D.R
   // RAP간 중복, 포함관계에 따라 추가, 삭제
      if (n'∈N(rap))
         Ri = Role_by_rap(rap)
         add_node(D.RAP(Ri r'), n')
         delete_node(D.RAP(Ri), n')
         delete_node(D.RAP(r'), n')
      fi
    rof
  fi
end
```

생성될 노드와 관련한 사이트 식별자 *s*, 역할 *r'*, 생성될 노드에 적용가능한 연산 *o*, 노드에 저장될 정보 *info*를 매개변수로 하여 생성될 노드의 연산도메인을 구한 후 노드를 생성하기 위해 현재 역할세션내의 역할 *r*이 도메인 보안관리자 역할 권한을 지녀야 한다. 먼저, 노드가 추가될 *R(r')* 존재여부에 따라 새로운 *R(r')*을 생성하고 정보구조표현층 내의 노드생성함수를 통해 노드를 생성한 뒤 현재 도메인내의 *RAP(r')*에 접근제어정보를 추가 한다. *RAP*내의 노드의 중복 및 포함관계에 따라 *RAP*간의 관계를 재구성한다.

■ 링크생성 알고리즘

```
Algorithm create_link(s, r', n, n')
begin
    // n, n' : 링크로 연결될 노드쌍
    // rsc = (s, r, n, t)
    D = DOMAIN(t)
       // 생성될 링크의 연산도메인을 구함
    if (r∈D.DSA)
    if (n∈D.N && n'∈D.N) then
      if RAPi(Role_by_node(n)) ==
        RAPi(Role_by_node(n')) then
        // 두 노드가 동일 RAP상에 포함
      else
        // 다른 RAP내의 노드로 링크생성
      fi
```

```
    else
        // site 간 링크생성
    fi
  fi
end
```

생성될 노드와 관련한 사이트 식별자 s, 역할 r', 링크로 연결될 노드쌍 n, n'를 매개변수로 하여 생성될 링크의 연산도메인을 구한 후 두 노드가 속한 사이트 및 RAP간의 위치에 따라 정보구조표현층내의 노드간 링크 연결함수를 통해 IRL, ERL, ISL를 생성한다.

4.5 시스템 보안관리자

시스템 보안관리자는 다음의 기능을 수행한다.

- 역할 및 역할계층구조 정의
- 도메인 생성, 삭제
- 생성된 도메인에서 실행가능한 연산의 지정

시스템 보안관리자는 도메인 보안관리자 역할을 정의하고 특정 사용자에게 역할권한을 부여 하므로서 도메인 보안관리자의 지정, 변경기능을 수행한다. 또한 시스템 보안관리자는 원격 시스템에 저장된 하이퍼텍스트 정보접근을 위하여 원격 시스템의 보안관리자와의 보안관리정보 교환기능을 담당한다. 시스템 보안관리자들은 원격 시스템에 저장된 하이퍼텍스트 정보의 노드에 대한 지역 사용자의 정보접근 허가를 요청하는 등록요청과 그에 대한 허가여부를 회신하는 결과회신 메시지를 통하여 보안관리정보를 교환한다.

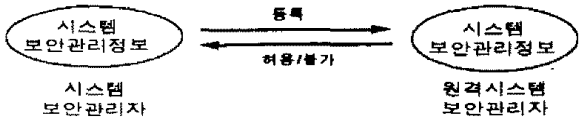
시스템 보안관리자는 관리대상 시스템내에 생성, 운영중인 도메인들에 대한 정보 및 원격 시스템 역할에게 허용된 도메인 정보를 저장하는 시스템 보안관리정보를 유지하는 기능을 수행한다. 시스템 보안관리자에 의해 유지, 관리되는 시스템 보안관리정보(SMIB:Security Management Information Base)의 구성은 다음과 같다.

$SMIB = \langle SSA, SR, DR, RAR \rangle$
SSA(System Security Administrator):시스템보안 관리자
SR(System Relationship):시스템간의 관계정보
DR(Domain Relationship):도메인간의 관계정보
RAR(Remote Accessible Request):원격 역할 요구 허용

위에서 원격역할 요구허용 정보단위는 (원격 호스트 ID, 원격역할, 노드, 연산)으로 구성된다.

시스템 보안관리자 사이의 등록을 위한 과정 및 등록요청에 포함된 정보는 (그림 4.4)과 같다.

Enroll = $\langle RH, RN, LH, LR, O \rangle$
RH(Remote Host):원격 시스템 ID
RN(Remote Node):원격 시스템내의 노드 ID
LH(Local Host):지역 시스템 ID
LR(Local Role):지역 시스템 역할
O(Operation):원격시스템 노드에 실행될 연산



(그림 4.4) 시스템 보안관리자간 등록
(Fig. 4.4) Enrollment of inter system security administrator

4.6 원격사용자 권한부여 대행자

시스템 보안관리자에 의해 관리되는 시스템 보안관리정보를 참조하여, 원격 사용자로부터 전송된 하이퍼텍스트 정보에 대한 접근권한 부여기능을 수행한다. 원격사용자 권한부여 대행자는 원격 시스템과 연결된 하이퍼텍스트 사용자들의 정보접근 요구에 대해 시스템 보안관리자들간의 메시지 교환을 통해 설정된 시스템 보안관리정보와 비교하여 그 허용여부를 결정한다.

4.5절에서 정의된 원격시스템 rh로부터의 원격역할 rr의 노드 n에 대한 연산 o의 접근요구에 대한 처리과정은 다음과 같다.

Algorithm *check_register*(rh, rr, n, o)
  // 해당 원격시스템의 권한 부여 대행자 호출
  // 권한 부여 대행자는 SMIB내에 요청된 접근 요구의 등록여부 확인
  // 이때 smi ∈ $SMIB$
if (rh = smi.원격호스트 ID and rr = smi.원격역할 and n = smi.노드 and o = smi.연산)

then return *access_granted* ;
else return *access_denied* ;
fi

4.7 접근제어

DHTS에서의 정보접근 요구는 한 시스템 내부에 저장된 정보에 대한 접근요구와 원격 시스템으로부터 전달된 접근요구가 있다.

역할 r이 현재 노드 n의 정보를 검색 또는 수정중일 때, 링크 l에 의해 n에 연결된 노드 n'에 대한 접근권한 여부를 결정하는 알고리즘은 다음과 같다.

```
Algorithm check_access_permission(rsc, n', o)
begin
  // rsc : 역할 세션 문맥
  // n' : 링크 l에 의해 노드 n에 연결된 검색
        방향 목적노드
  // o : 노드 n'에 적용할 연산

  do current_D in DOMAIN(o)
    // 노드 n과 n'이 동일한 RAP(Ri) 내에
      포함되어 있음
    if LT(l) ∈ current_D.IRL then
      return access_granted ;
    // 노드 n과 n'이 동일한 RAP(Ri)에 포함
      되어 있지 않은 경우
    else if LT(l) ∈ current_D.ERL then
    // 역할 계층에 따라 접근허용/불가 결정
      if r ≥ Role_by_node(n') then
      return access_granted ;
      else
      return access_denied ;
    // n'가 다른 사이트내에 존재할 경우
    else if LT(l) ∈ current_D.ISL then
      return(check_register(s, r, n', o))
      // 접근하고자 하는 원격 시스템내
        SMIB.RAR내에 등록여부 확인
    fi
  od
end
```

도메인의 정의와 그 구성요소에서 알 수 있듯이 지역역할의 접근요구에 대한 접근제어는 역할기반 접근제어 모델이 적용된다. 지역시스템내의 정보에 대한 접근요구시 역할이 현재 접근중인 *RAP* 내에 목적노드가 위치해 있을 경우 접근권한을 부여하며, 원시노드와 목적노드가 서로다른 *RAP* 상에 위치할 경우에는 역할간 관계에 따라 접근권한 부여 여부를 결정한다.

원격역할로 부터의 하이퍼텍스트 접근요구는 지역시스템에서 관리하는 보안관리정보내에 해당 원격역할의 정보가 포함되어 있는 지를 확인하는 과정을 통해 그 허용여부가 결정된다. 사용자에게 부여된 역할 r의 역할세션 문맥 rsc가 (s, r, n, t)일 때, 노드 n에 연결된 n'노드의 정보검색 및 수정 알고리즘은 다음과 같다.

```
Algorithm process_info_of_node(n', o)
begin
    // rsc : 역할세션 문맥
    // n : 현재 정보검색 또는 수정중인 노드
    // n' : 링크 l에 의해 노드 n에 연결된 노드
    // t : 노드 n의 도메인타입
    // o : 실행연산 종류
    if check_access_permission(rsc, n', o) ==
      access_granted then
        t' = DOT_by_node(n');
        rsc = (s, r, n', t');
        if o == v then // 검색 연산
            // 노드 n'의 정보를 사용자에게 정보
              표현계층 서비스를 통해 제공
        else // 수정 연산
            // 사용자에게 노드 n'의 정보수정
              서비스를 제공
        fi
    else //access_denied
        return ;
  fi
end
```

4.8 모델 비교 및 분석

접근제어정책과 관련하여 기존에 연구되어진 DHTS 보안모델[2, 8, 9, 16]과 본 논문에서 제안한 보안모델의 특성을 비교하면 다음 〈표 4.1〉과 같다.

〈표 4.1〉 보안모델들의 특성 비교
〈Table 4.1〉 Compare of security model

보안 모델 / 특징	P. Samarati & E. Bertino 모델	S. DeRose 모델, V. Parunak & M. Olivier 모델	제안 모델
적용 보안 정책	자율적 접근제어	강제적 접근제어	역할기반 접근제어
사용자 접근권한	사용자	사용자 등급	사용자 그룹
접근방법	모든 접근요구	제한적(읽기/쓰기)	모든 접근요구
접근권한 부여방법	소유권 또는 보안관리자	중앙집중 보안관리자	사이트별 시스템 보안관리자
보안등급	사용안됨	각 접근대상 정보 (노드 및 링크)	각 접근대상 정보에 역할계층에 따른 권한등급 부여
장점	융통성	- 노드와 링크의 라벨링 - 링크의은폐 및 대체 정보에 의한 보안의 고신뢰성	- 안정성 및 효율적 권한 관리 - 접근제어 판단 부담 줄임
단점	- 저신뢰성 - 접근제어정보 저장 및 갱신 부담 - 접근제어판단 부담	- 매우 낮은 융통성 - 접근대상정보에 보안등급 부여가 선험적	- 접근방법(연산 도메인)의 증가시 노드의 중복 부담 - 역할 계승, 사상에 따른 복잡성
적용분야	엄격한 보안 환경이 요구되지 않는 응용분야	특정영역에 선택적 적용	접근대상 정보에 대한 접근방법이 단순한 상업적 응용분야

일반적으로 자율적 접근제어를 기반으로 한 보안모델이 HTS상에 가장 적합할 것으로 여겨지고 있으며 상대적으로 많은 응용분야에 걸쳐 적용되고 있지만 위에서 언급한 구조적 부담이 있으며, 강제적 접근제어를 기반으로 한 보안모델에서는 보안대상 객체에 보안등급을 부여하기 위해 노드와 링크의 분류가 선행되어야 하는데 표준화된 분류 자체가 불가능하기 때문에 특정 응용분야에 적합한 제한적 분류나 선험적(heuristic) 분류에 의존하고 있으며 광범위한 적용을 위해서는 추후 더욱 세부적이고 일반화된 분류가 요구된다.

5. 결 론

본 논문은 DHTS 보안모델을 설계하기 위해 HTS의 특성 및 보안 메카니즘을 고찰하고, 다양한 접근제어 정책들의 HTS상에 적용시 문제점을 제시하였으며 본 모델의 설계시 필요한 구성요소 및 개념을 정의하고 도메인 계층상에 역할접근제어를 기반으로 한 보안모델을 제안하였다.

본 논문에서 제안된 연산도메인은 도메인내에 포함된 모든 노드에 적용되는 연산이 동일하고, 연산도메인내의 각 노드들은 관련된 노드 및 링크의 집합인 *RAP*으로 구성된다. 제안된 연산도메인에 역할기반 접근제어를 적용한 보안모델의 장점으로 첫째, 시스템상에 역할기반 접근제어를 적용하므로서 사용자 단위로 접근허용 권한을 부여하는 자율적 접근제어를 기반으로 한 보안모델보다 더욱 안전한 접근제어가 가능하며 둘째, 사용자는 역할에 따라 *RAP*내의 모든 노드를 추가의 접근권한여부 확인 절차없이 접근가능함에 따라 접근제어 판단에 따른 부담을 줄였으며 셋째, 사용자와 접근대상 정보에 대한 접근권한을 할당하는 권한부여를 논리적으로 분할하므로서 노드 및 사용자 증가에 따른 정보저장 및 갱신시 모든 사이트마다의 역할 접근제어정보를 수정해야 하는 부담을 줄이므로서 효율적 권한관리를 가능케 하였다. 반면 노드에 실행가능한 연산의 종류수 만큼 각 연산도메인내에 노드 및 링크가 중복 구성되므로서 최대 연산도메인 수 * 노드의 수 만큼의 기억공간이 필요하게 되고, 노드를 포함하는 사이트간 항해패턴에 따라 링크의 타입 비교를 위한 처리시간 부담이 증가될 수 있다.

추후 연구과제로서 원격사이트내의 정보 접근시 사이트간 통신을 위한 프로토콜의 상세한 정의와 상이한 역할계층 구조를 지닌 사이트간 역할사상(inter

site role-mapping) 알고리즘에 관한 연구가 필요하다.

참 고 문 헌

[1] R.M. Akscyn, D.L. McCracken, "KMS : A distributed hypermedia system for managing knowledge in organizations," Comm. of the ACM, July 1988.

[2] E. Bertino, F. Origgi, P. Samarati, "An Extended Authorization Model for Object Databases," J. Computer Security, vol. 3, no. 6, pp. 169-206, 1995.

[3] B. Campbell, J.M. Goodman, "HAM : A general purpose hypertext abstract machine," Comm. of the ACM, July 1988.

[4] E.B. Fernandez, K.R. Nair, "High-Level Security Issues in Multimedia/Hypertext Systems," Proc. of IFIP, 1996.

[5] F. Halasz, M. Schwartz, "The Dexter Hypertext Model," Comm. of the ACM, 1994.

[6] J. Kahan, "A capability-based authorization model for the World-Wide Web," Computer Networks & ISDN Systems 27, 1995.

[7] M. Nyanchama, "Modeling MAC in Role Based Security Systems," IFIP Confer. on Database Security, Aug. 1995.

[8] M.S. Olivier, "A Heuristic for Securing Hypertext Systems," IFIP, 1996.

[9] P. Samarati, E. Bertino, "An Authorization Model for a Distributed Hypertext System," IEEE Transactions on knowledge and data eng. vol. 8. No. 4. Aug. 1996.

[10] B. Thurasingham, "Multilevel Security for Information Retrieval System," Information and Management, 1995.

[11] J.K. Millen, "Models of multilevel computer security," in Advances in Computers M.C.Yovits (Ed.), vol. 29, Academic Press., 1989.

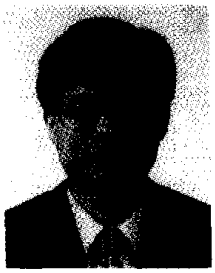
[12] E.B. Fernandez, E. Gudes, H. Song, "A model for evaluation and administration of security in object-oriented databases," IEEE Trans. on Data and Knowledge Engineering, vol. 6,, no. 2, April 1994, 275-292.

[13] C. Wood and E.B. Fernandez, "Authorization in a decentralized database system," Proc. 5th Int. Conf. on Very Large Databases, 1979, 352-359.

[14] R. Sandhu, E. Coyne, H. Feinstein, C. Youman, "Role-Based Access Control models," IEEE Computer, vol. 29, no. 2, Feb. 1996, 38-47.

[15] M. Nyanchama and S.L. Osborn. "Access Rights Administration in Role-Based Security Systems," In J. Biskup, M. Morgenstern, and C. Landwehr, editors, Database Security VIII : Status Prospects, pages 37-56. North-Holland, August 1994.

[16] S.J. DeRose, "Expanding the Notion of Links," Hypertext'89 Proceedings.



정 철 윤

1987년 전남대학교 전산학과 졸업(이학사)

1989년 한국외대 경영정보대학원 전산학과 졸업(이학석사)

1989년~1992년 삼보컴퓨터 기술연구소

1993년~현재 광주여자대학교 전산학과 전임강사

1994년~현재 전남대학교 전산학과 박사과정

관심분야 : 정보보안, 컴퓨터 네트워크, 멀티미디어 시스템 등

이 형 효

1987년 전남대학교 전산학과 졸업(이학사)

1989년 한국과학기술원 전산학과 졸업(공학석사)

1990년~1992년 삼보컴퓨터 기술연구소

1993년~1997년 한국통신 연구개발원

1995년 정보처리기술사(전자계산조직응용)

1997년~현재 전남대학교 전산학과 박사과정

관심분야 : 통신망관리, 정보보안, 객체지향시스템 등

노 봉 남

1978년 전남대학교 수학교육과 졸업(이학사)
1982년 한국과학기술원 전산학과(공학석사)
1994년 전북대학교 대학원 전산통계학과(이학박사)
1983년~현재 전남대학교 전산학과 교수

관심분야: 객체지향시스템, 통신망관리, 정보보안, 컴퓨터와 정보사회 등