情報保護學會論文誌
第16卷 第3號, 2006. 6

사용목적 분류를 통한 프라이버시 보호를 위한 접근제어 모델

나 석 현†, 박 석‡
서강대학교

An Access Control Model for Privacy Protection using Purpose Classification

Seok-Hyun Na†, Seog-Park‡
Sogang University

요 약

사용목적(Purpose)은 최근 개인 프라이버시 보호와 관련하여 데이타 수집과 수집 후 보안관리에 있어서 중요한 요소로 사용되고 있다. W3C(World Wide Web Consortium)는 데이타 제공자가 자신이 방문한 웹 사이트에 개인정보를 제공하는 것을 통제할 수 있도록 하는 표준을 제시하였다. 그러나 데이타 수집 후 유통과정에서 개인정보에 대한 보안관리에 대한 언급이 없다. 현재 히포크라테스 데이타베이스(Hippocratic Databases), 사용목적기반 접근제어(Purpose Based Access Control)등은 W3C의 데이타 수집 메커니즘을 따르고 있으며, 데이타 수집 후 보안관리에 대하여 사용목적 관리와 접근제어 기법을 사용하여 관리를 하고 있으나 사용목적에 대한 표현과 사용목적 관리의 미흡함으로 인하여 그에 따르는 개인정보의 프라이버시 보호에 있어서 효과적인 해결책을 제시하지 못하고 있다.

본 논문은 사용목적의 표현력을 향상시키면서, 사용목적의 효과적인 관리기법을 제시한다. 또한 개인의 프라이버시 보호를 위한 방법으로 사용목적의 분류를 통해 최소권한의 원칙을 따르는 접근제어 기법을 제시한다. 본 논문에서는 사용목적을 상속적, 시간적 그리고 독립적 구조로 분류하였으며, 이렇게 분류된 사용목적에 대한 각기 다른 관리 기법을 제시한다. 또한 접근제어의 유연성을 위해 RBAC의 역할계층 구조를 사용하였으며, 일의 최소 단위인 태스크(task)의 최소권한을 얻기 위한 조건으로 몇몇 특성의 사용목적을 사용하여 만족할 경우 태스크를 처리하기 위한 기존 모델보다 향상된 최소사용권한을 제공하는 기법을 제시한다.

ABSTRACT

Recently purpose is used by an crucial part to security management when collecting data about privacy. The W3C(World Wide Web Consortium) describes a standard spec to control personal data that is provided by data providers who visit the web site. But they don't say anymore about security management about personal data in transit after data collection. Recently several researches, such as Hippocratic Databases, Purpose Based Access Control and Hippocratic XML Databases, are dealing with security management using purpose concept and access

접수일: 2006년 1월 23일; 채택일: 2006년 6월 1일
† 주저자, shna@sogang.ac.kr
‡ 교신저자 : spark@sogang.ac.kr

control mechanism after data collection a W3C's standard spec about data collection mechanism but they couldn't suggest an efficient mechanism for privacy protection about personal data because they couldn't represent purpose expression and management of purposes sufficiently.

In this paper we suggest a mechanism to improve the purpose expression. And then we suggest an accesscontrol mechanism that is under least privilege principle using the purpose classification for privacy protection. We classify purpose into Along purpose structure, Inheritance purpose structure and Stream purpose structure. We suggest different mechanisms to deal with them. We use the role hierarchy structure of RBAC(Role-Based Access Control) for flexibility about access control and suggest mechanisms that provide the least privilege for processing the task in case that is satisfying using several features of purpose to get least privilege of a task that is a unit of business process.

Keywords : *Hippocratic Databases, Purpose, Task, Privacy, RBAC*

Ⅰ. 서 론

최근 들어 사적인 데이타가 인터넷을 통하여 데이타베이스에 점점 더 많이 저장되고 있으며, 이렇게 저장된 사적인 데이타가 개인과 기업 그리고 기업들 간에 유통이 됨으로써 인해 개인 프라이버시(privacy)의 민감한 문제로 대두되고 있다.

현재 개인의 프라이버시 보호와 관련하여 많은 관심을 가지고 활발한 연구가 수행되어 지고 있으며, 그 예로 W3C에 의해 표준으로 제안되어진 P3P (Platform for Privacy Preference) [1]와 IBM에 의해 제안되어진 EPAL(Enterprise Privacy Authorization Language) [2]이 있다. 이 표준들은 데이타 수집 과정에서 개인의 프라이버시 보호를 위해 프라이버시 정책(privacy policy)과 프라이버시 선호(privacy preference)를 명시하고 비교하는 메카니즘을 제시하였다. 이중 데이타 수집과정에서 프라이버시 정책과 프라이버시 선호에 사용되는 요소 중 중요하게 사용되는 요소가 사용목적이다.

그러나 P3P와 EPAL은 데이타의 수집과 사용방법을 제시하였으나, 데이타 수집 후 유통과정에서 보안관리에 대한 언급이 없다. IBM의 히포크라테스 데이타베이스(Hippocratic Databases) [3]는 P3P와 EPAL의 데이타 수집 메카니즘을 이용하며 또한 보안관리기법을 적용한 것이다. 그 이외에 Purdue University의 사용목적기반 접근제어(Purpose Based Access Control) [4], 이재길의 히포크라테스 XML 데이타베이스(Hippocratic XML Databases) [5], IBM의 고객 데이터의 privacy-enabled management (Privacy-enabled Manageme nt of Customer Data) [6]등이 연구되고 있으며, 이들 연구는 데이타 수집 후 보안관리에 대해 사용목적과 그에 따른 접근제어를 통하여 이루어진다. 그러나 이들 연구에서 제시하는 사용목적의 관리는 사용목적 특성에 따르는 표현력의 부족으로 인하여 관리에 어려움이 있으며, 이로 인해 개인정보 제공에 대한 최소권한의 원칙을 만족하지 못한다.

본 논문에서는 사용목적을 다각적인 면에서의 분석을 통하여 상속적, 시간적 그리고 독립적 구조로 분류하였으며, 이렇게 분류된 사용목적들에 대한 접근제어 기법을 제시한다. 또한 접근제어의 유연성을 위해 RBAC [7]의 역할계층 구조를 사용하였으며, 일의 단위인 태스크의 최소권한을 얻기 위한 조건으로 위에서 언급된 몇몇 특성의 사용목적을 사용하여 만족할 경우 태스크를 처리하기 위한 기존 모델보다 향상된 최소의 사용권한을 제공하는 기법을 제시한다.

우선 개인정보 보안관리를 위한 연구들로 즉, 히포크라테스 데이타베이스, 사용목적기반 접근제어 모델 등에 대해 살펴보며, 또한 논문의 동기인 사용목적 관리와 접근제어의 분석을 통한 문제점을 제시한다. 이어서 사용목적 분류를 통한 문제점 해결과 보다 강화된 최소권한의 원칙을 만족하는 접근제어 기법을 제시한다. 그리고 기존의 사용목적을 이용한 접근제어 기법과의 장단점을 실험을 통하여 분석한다.

Ⅱ. 관련연구 및 문제점

여기서는 개인의 프라이버시 보호를 위한 기존의 모델과 검사 방법 그리고 기존의 접근제어 방법 중 본 논문의 이해에 필요한 접근제어 기법에 대하여 알아보고, 접근제어 관점에서 접근할 경우 기존의 모델의 문제점에 대하여 살펴본다.

purpose	table	attribute	authorized-user
purchase	customer	email	{shipping, customer-service}
purchase	customer	credit-card-info	{charge}
registration	customer	name	{registration}
...	...	...	...

(a) 프라이버시-권한 테이블

purpose	customer-id	name	email	...
{purchase, registration}	0	Bob	bob@ibm.com	...
{purchase}	1	Alice	alice@microsoft.com	...
{registration}	2	Mallory	mallory@hotmail.com	...
{registration, purchase-circles}	3	Trent	trent@oracle.com	...
...	...	...	...	...

(b) 데이타 테이블 : 고객 테이블

그림 1. 히포크라테스 데이타베이스 스키마의 예

1. 히포크라테스 데이타베이스

히포크라테스 데이타베이스[3]는 관계형 데이타베이스에 프라이버시 보호 기능을 추가한 데이타베이스 모델이다. 히포크라테스 데이타베이스 스키마의 예와 질의 처리 방법을 설명하면 다음과 같다. 그림 1은 히포크라테스 데이타타베이스의 스키마의 예를 나타낸다.

프라이버시 권한은 데이타의 사용목적, 테이블 이름, 속성 이름, 허가된 사용자를 포함하며, 그림 1(a)와 같이 프라이버시-권한 테이블에 저장된다. 데이타 레코드의 사용목적은 그림 1(b)와 같이 데이타 테이블의 purpose 속성에 저장된다.

질의 처리 과정에서 히포크라테스 데이타베이스는 속성 접근제어(attribute access control)와 레코드 접근제어(*record access control*)를 연속해서 수행한다. 질의에는 질의를 수행하고자 하는 사용목적이 함께 명시되며, 이는 속성 접근제어와 레코드 접근제어에 사용된다. 속성 접근제어는 질의에 명시된 테이블의 속성을 질의 사용목적을 위해 액세스할 수 있도록 허용하는 프라이버시 권한이 부여되어 있는지를 검사하는 것이다. 레코드 접근제어는 액세스하려는 레코드에 저장된 purpose 속성의 값이 질의 사용목적과 일치하는지를 검사하는 것이다. 이는 일반적인 관계형 데이타베이스 시스템에서 수행하지 않는 과정으로, 히포크라테스 데이타베이스에서만 추가적으로 수행하는 과정이다.

예를 들어 그림 1에서 customer-service 라는 사용자가 purchase 사용목적을 위해 고객 테이블에서 e-mail을 요구하는 질의를 수행한다고 가정한다. 첫째, 속성 접근제어에서 customer-service 사용자가 purchase 사용목적을 위해 고객 테이블의 email 속성을 액세스할 수 있는지 검사한다. 이 경우, 프라이버시-권한 테이블의 첫 번째 프라이버시 권한이 그 사용자에게 액세스를 허용하도록 부여되어 있다. 둘째, 레코드 접근제어에서는 액세스하는 레코드의 purpose 속성에 purpose 사용목적이 포함되어 있는지 검사한다. 이 경우, 고객 테이블의 첫 번째와 두 번째 레코드의 purpose 속성에는 purpose 사용목적이 포함되어 있으므로, 질의 결과로 Bob과 Alice의 e-mail 이 반환된다.

2. 사용목적기반 접근제어

사용목적기반 접근제어 모델[4]은 사용목적 관리를 위한 사용목적 트리와 접근제어를 위해 RBAC의 역할계층 구조를 이용한다.

사용목적기반 접근제어는 하나의 계층구조로서 사용목적을 관리한다. 상위 사용목적은 하위 사용목적을 일반화(generalization)하며, 하위 사용목적은 상위 사용목적을 특수화(specialization)한다. 다음의 그림 2는 사용목적 트리를 나타낸다.

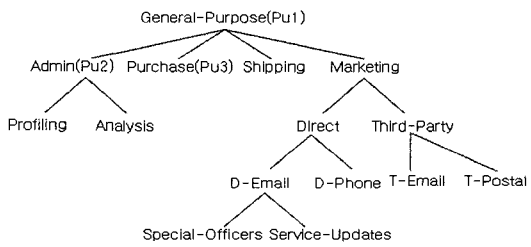


그림 2. 사용목적 트리의 예

사용목적기반 접근제어는 기업환경에서의 접근제어를 위해 RBAC의 역할구조를 기반으로 APA (Access Purpose Authorization)를 사용하여 사용자에게 권한을 부여한다. APA는 〈ap, cr〉의 2-튜플로 구성되어지며, cr(conditional roles)은 〈r, C〉의 2-튜플로 구성되어 있는데, r(role)은 역할을 의미하며 C(Conditional Role)은 역할 속성(role attributes)과 시스템 속성(system attributes)을 의미하며, 이것은 역할이 사용목적을 활성화하기 위한 조건으로 사용된다. 이때 C는 역할에

할당되며, 상위로부터 하위로 상속의 속성을 지니고 있다. 그림 3는 사용목적을 표현한 역할계층 구조와 사용목적을 활성화하기 위한 조건인 C를 표현한 계층구조이다.

예를 들어 CR ={CanUpdate}이 〈(ExpLevel 〉 5) ∧ (ServiceType = "Update-Info")〉라 가정한다. 이때 역할 CanUpdate에 속하는 사용자 u가 ExpLevel = 7 이고 ServiceType = "Updte-Info"라 할 경우

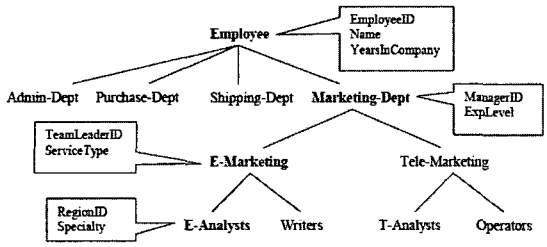


그림 3. Role Hierarchy 와 Role Attribute 의 예

사용자 u는 "E-Marketing" 역할을 활성화 할 수 있다. 이때 이에 해당하는 사용목적을 활성화하여 사용자가 요구한 사용목적의 사용권한을 부여 받을 수 있다.

사용자가 u가 역할 CanUpdate에 속하며, ExpLevel 〈 7 또는 ServiceType ≠ "Updte-Info"라 할 경우 사용자 u는 "E-Marketing" 역할을 활성화 할 수 없다. 또한 사용자가 요구한 사용목적의 사용권한을 부여 받을 수 없다.

3. 히포크라테스 XML 데이타베이스

관계형 데이타베이스는 평면(flat) 구조를 가지는 반면 XML 데이타베이스는 트리 형태의 계층 구조를 가진다. 이러한 구조적인 차이점으로 인해 프라이버시 선호 및 정책, 프라이버시 권한, 데이타 레코드의 사용목적과 같은 히포크라테스 데이타베이스 모델에서 사용하는 프라이버시 정보들이 XML 데이타 트리의 임의의 엘리먼트에 명시될 수 있도록 확장되어야 한다. 히포크라테스 XML 데이타베이스 모델[5]은 프라이버시 선호 및 프라이버시 정책 모델과 프라이버시 권한 모델로 구성된다.

히포크라테스 XML 데이타베이스 모델은 데이타 구조를 XML 데이타의 특성인 트리 형태의 구조와 하나의 DGA[3]형태인 계층구조로 사용목적을 확장하여 표현한다.

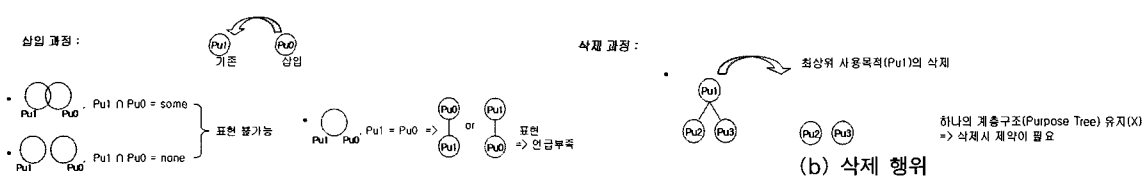
히포크라테스 XML 데이타베이스 모델은 히포크라테스 데이타베이스 모델을 XML 환경으로 확장한 것이며, 사용목적을 DGA형태의 계층구조를 사용하여 표현하였다.

4. 문제점

4.1. 사용목적 관리 측면

사용목적기반 접근제어와 히포크라테스 XML 데이타베이스 모델은 히포크라테스 데이타베이스 모델의 문제점인 사용목적의 중복성과 객체의 삽입·삭제 행위시 비일관성 문제를 해결하며, 다음과 같은 특성을 지닌다. 사용목적을 각각 내포관계의 특성을 고려한 하나의 내포관계 특성과 DGA 계층구조 형식을 고려한 하나의 사용목적 트리 형식으로 분류·관리 한다. 그러나 이 두 가지 모델의 사용목적 계층구조 관리 방식으로 사용목적을 관리할 경우, 위에서 언급한 사용목적의 특성 그리고 계층구조의 특성인 최상위 노드를 제외한 모든 노드들은 부모 노드를 가져야 한다는 제약 사항으로 인하여 문제점이 발생한다. 다음 그림 4는 사용목적의 추가·삭제의 행위시 4가지 상황에서 사용목적의 표현력과 사용목적 관리의 미흡함으로 발생하는 문제점을 보인다. 삽입·삭제 행위에서 비교대상이 되는 것은 권한의 집합이다.

그림 4(a)은 기존 사용목적 Pu1에 Pu0을 삽입시 이 둘간의 권한 중 일부가 동일한 경우, 동일한 것이 없는 경우 그리고 동일한 경우에 기존의 사용목적 관



(a) 삽입 행위 (b) 삭제 행위

그림 4. 기존-사용목적의 삽입 삭제 행위시 문제

리 기법의 표현력 부족으로 해결이 안됨을 보여준다.

그림 4(b)은 기존의 사용목적 트리에서 최상위 사용목적 Pu1을 삭제시 하나의 계층구조로 표현이 불가능하며, 또한 계층구조의 특성인 하위 노드는 상위 노드를 가져야 하는 제약을 벗어남으로 인하여 문제가 발생한다.

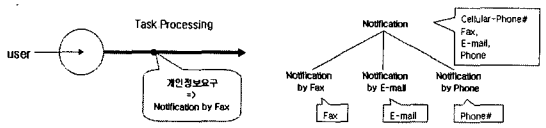
이들 문제를 사용목적 표현력의 미흡함을 첫 번째 문제점으로 하며, 삽입·삭제시 사용목적 관리의 미흡함을 두 번째 문제점으로 정의한다.

4.2. 접근제어 측면

위에서 설명한 사용목적기반 접근제어와 히포크라테스 XML 데이타베이스 모델의 접근제어 메커니즘을 보면 다음과 같은 상황에서 문제점이 발생한다.

상황은 사용자가 현재 Notification, Notification by Fax, Notification by E-mail, Notification by Phone# 의 4가지의 사용목적을 활성화 시킬 수 있는 경우이며, 이때 태스크 프로세스상에서 사용목적 Notification by E-mail의 권한인 데이타 제공자의 E-mail 정보를 요구하는 상황이다. 직관적으로 보아도 현 상황에서 사용자에게 E-mail 정보만을 허용하는 것이 최소권한의 원칙을 따르는 것임을 알 수 있다.

이때 사용목적기반 접근제어 모델의 메커니즘으로 접근제어를 하는 경우를 생각해보자. 위에서 설명했듯이 사용자는 그림 5(b)에서 보듯이 4가지 사용목적을 활성화 시킬 수 있는 상황이다. 이때 사용자가 사용목적 Notification by E-mail의 사용 목적이 아닌 그보다 상위의 사용목적 Notification을 사용하여 권한을 요청할 경우 E-mail 이외의 권한인 Celluler-Phone#, Fax, Phone#의 개인정보까지도 제공되므로 최소권한의 원칙이 위배된다.



(a) 상황 (b) 사용목적 계층구조와 권한

그림 5. 기존-접근제어 상황

히포크라테스 XML 데이타베이스 모델의 메커니즘으로 접근제어를 하는 경우에도 같은 결과로 최소권한의 원칙이 위배되는 문제점을 들어내며, 이를 세 번째 문제점으로 정의한다.

본 논문에서는 위에서 언급한 사용목적 관리시 발생하는 사용목적 표현력의 미흡함, 삽입·삭제시 사용목적 관리의 미흡함과 접근제어시 발생하는 최소권한의 원칙 위배의 문제점을 각각 3-타입의 사용목적 분류와 Pu-ARBAC모델을 제시함으로써 해결하려고 한다.

Ⅲ. 사용목적 분류를 통한 보안 접근제어 기법

1. 사용목적 분류 및 관리

사용목적 분류의 목적은 개인의 프라이버시 보호를 위해 사용자에게 개인정보를 제공하는 단위인 사용목적을 보다 현실세계에 맞게 분류함으로서 기존의 모델보다 향상된 최소권한의 원칙을 적용함에 있다. 또한 이들 사용목적의 관리를 통하여 각기 다른 특성을 지닌 사용목적들을 일관적으로 관리함으로써 접근제어에 보다 편리함을 제공한다.

1.1. 사용목적 분류

본 절에서는 사용목적을 대상으로 각기 다른 특성으로 상속적, 시간적, 독립적 구조의 3가지 구조로 분류한다.

정의 1) 사용목적 트리(Purpose Tree)를 PT라 하고, Pu는 PT 의 모든 사용목적들 중 어느 한 사용목적(Purpose)이라 하며, 그리고 P는 개인정보에 관한 권한(Permission)의 집합이라 하자. 이때 P_i는 집합 P 중 한 원소가 되며, $Pu_i = \{P_1, P_2, \ldots, P_n\}$ 이다.

정의 2) 모든 사용목적(Purposes)들은 3-타입 중 하나로 분류되어지며, 3-타입의 분류는 다음과 같다.

가) Inheritance Purpose(타입 I): 상속적구조의 사용목적
$\forall i$, purpose $Pu_i \in$ type I $\Rightarrow$
$Pu_i \in$ set of inheritance purpose
나) Stream Purpose(타입 S): 시간적 구조의 사용목적
$\forall i$, purpose $Pu_i \in$ type S $\Rightarrow$
$Pu_i \in$ set of stream purpose
다) Alone purpose(타입 A): 독립적 구조의 사용목적
$\forall i$, purpose $Pu_i \in$ type A $\Rightarrow$
$Pu_i \notin$ set of inheritance purpose
$\wedge$ $Pu_i \notin$ set of stream purpose
으로 분류하며,
$\forall i$, purpose $Pu_i \in$ type A and $Pu_i \in$ type S $\Rightarrow$ $Pu_i \in$ set of inheritance purpose $\wedge$ $Pu_i \in$ set of stream purpose
이 존재 가능하다.

정의 3) PT를 사용목적 트리(Purpose Tree)라 하고, Pu는 PT 의 모든 사용목적들 중 어느 한 사용목적(Purpose)이라 할 때, 사용목적 Pu는 타입에 따라 다음과 같은 규칙을 따른다.

타입 I : - 상위 사용목적은 하위사용목적을 내포한다.
타입 S : - 상위 사용목적을 활성화하기 위한 조건으로 하위 사용목적이 조건(시간적인 조건)이 만족 되어져야 하며, 하위 사용목적들은 AND, OR 관계로 이루어져야 한다.
타입 A : - 타입 I 와 타입 S 이외의 모든 사용목적들은 타입 A 로 구별되어 진다.

이들 분류 기준은 실제 비지니스 프로세스(business process)상에서 사용목적이 사용됨을 기반으로 하며, 이들 사용목적들 간에는 삽입·삭제의 행위 과정에 있어서 상이하게 다른 기준이 적용된다. 그로 인해 서로 다른 방식의 삽입·삭제의 행위가 발생된다. 삽입·삭제 행위시 본 논문에서는 너비우선 탐색을 가정하며, 정의 3의 규칙을 따라야 한다.

본 논문에서는 사용목적의 삽입·삭제 행위시 따라야 할 규칙(정의3)과 이들 3가지 특성으로 분류되어진 사용목적(타입 I, 타입 S, 타입 A)들이 현실에 존재하는 모든 사용목적들을 표현 가능해야 한다는 것을 요구사항으로 한다.

3.1.2 절에서는 사용목적 관리를 통하여 이들 요구사항들이 만족되어짐을 보이며, 그로 인해 2.4.1 절에서 제시한 문제점인 사용목적 표현력과 삽입·삭제시 사용목적 관리의 미흡함이 해결됨을 보인다.

1.2. 사용목적 관리

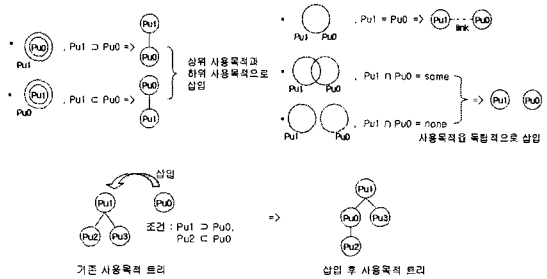
3.1.1 절에서 사용목적의 특성에 따라 3-타입으로 분류를 하였다. 이들 분류된 사용목적들 간에는 각각 특성에 따라 서로 다른 삽입·삭제의 행위가 일어나며, 삽입·삭제의 행위는 각각 사용목적의 특성에 따라 일관적으로 관리를 해주어야 한다.

• 타입 I

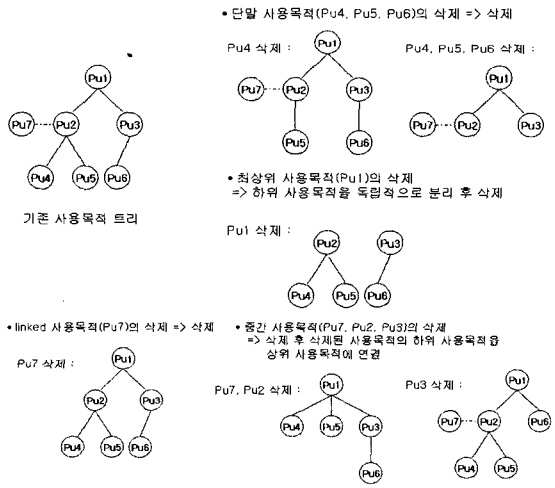
타입 I는 하위 사용목적에서 상위 사용목적으로의 권한 상속이 이루어지는 특성을 지니고 있으며, 사용목적 트리를 이룬다.

삽입·삭제의 행위 과정에서 input_purpose_typeI(Pu_i), delete_purpose_typeI(Pu_i)의 두 함수 알고리즘[부록A]이 사용되며, 이들 함수는 $Pu_i = \{P_1, P_2, \ldots, P_n\}$ 즉, 권한의 집합을 비교대상으로 한다. 다음 그림 6은 위의 삽입·삭제 알고리즘을 이용한 타입 I의 행위를 예를 들은 것이며, 사용목적의 표현과 삽입·삭제시 사용목적 관리의 미흡함으로 발생하는 문제점의 해결됨을 보인다.

삽입·삭제의 행위 과정에서 사용되는 함수는 사용목적 트리에 하나 이상의 사용목적이 존재하고 있음을 가정한다. 실선은 사용목적 간에 상속적 특성을 지닌 부모·자식 관계를 의미하며, 일점쇄선은 권한이 같은 사용목적을 표현한 링크를 의미한다.



(a) 사용목적 삽입예



(b) 사용목적 삭제예

그림 6. 타입 I - 사용목적 관리의 예

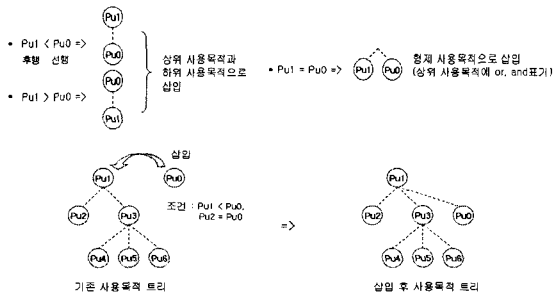
• 타입 S

정의 4) Pu는 PT 의 모든 사용목적들 중 어느 한 사용목적(Purpose)이라 하자. 이때 Pu_t는 집합 PT 중 시간적 특성을 지닌 사용목적이며, Pu_t 은 $Pu_1 \rightarrow Pu_2 \rightarrow \ldots \rightarrow Pu_n$ 와 같은 시간적(타입 S) 특성을 지닌다.

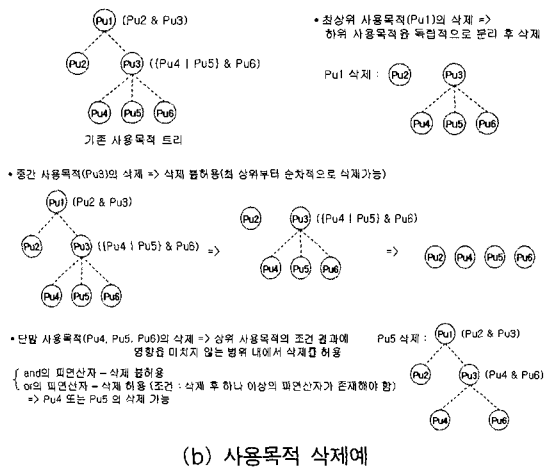
타입 S는 상위 사용목적을 활성화시키기 위한 조건으로 하위 사용목적이 만족 되어져야 하는 특징을 가지고 있으며, 사용목적 트리를 이룬다.

삽입·삭제의 행위 과정에서 input_purpose_typeS(Pu_t), delete_purpose_typeS(Pu_t)의 두 함수 알고리즘[부록A]이 사용되며, 이들 함수는 삽입·삭제 과정에 있어서 사용목적들 간의 선후를 비교 대상으로 한다. 이때 사용되는 선후 관계는 비즈니스 프로세스를 알고 있는 사용목적 관리자가 미리 명시하며, 또한 사용목적 관리자는 상위 사용목적의 하위에 여럿 사용목적들이 존재 할 경우 하위 사용목적들을 AND, OR관계를 기술해 주어야 한다. 다음 그림 7은 위의 삽입·삭제 알고리즘을 이용한 타입 S의 행위를 예를 들은 것이다.

삽입·삭제의 행위 과정에서 사용되는 함수는 사용목적 트리에 하나 이상의 사용목적 이 존재하고 있음을 가정한다. 이때 파선은 사용목적간에 시간적 특성을 지닌 부모·자식 관계를 의미한다.



(a) 사용목적 삽입예



(b) 사용목적 삭제예

그림 7. 타입 S - 사용목적 관리의 예

• 타입 A

타입 I 와 타입 S 로 표기가 안되는 모든 사용목적을 타입 A로 표기한다. 타입 A의 사용목적 삽입·삭제는 타입 I 와 타입 S의 삽입·삭제 함수 알고리즘에 포함되어 표기된다.

3-타입으로 표현된 사용목적은 xml 문서로 표기되어 유지관리 된다. 이때 xml 문서는 Purpose Language[부록B]의 유효성(validity)을 만족해야 한다.

그림 8은 정보제공자가 제공한 사용목적에 부합하는 권한을 사용목적 트리에 같이 표기한 예이다.

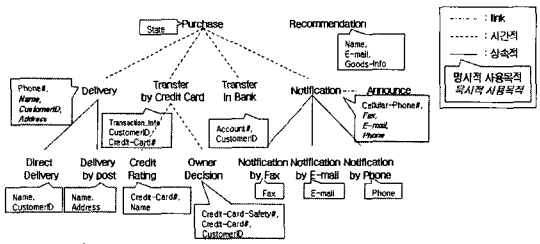


그림 8. 사용목적 트리-권한 부여의 예

3-타입의 분류에 따른 각각의 특성에 따라 제시한 알고리즘에 의해 사용목적을 관리함을 보였으며, 이 예를 통하여 사용목적 표현과 삽입·삭제시 사용목적 관리의 문제점이 해결됨을 보였다.

다음절에서는 이렇게 분류를 통해 관리되는 사용목적을 사용하는 보안 접근제어 모델인 Pu-ARBAC을 제시함으로써 접근제어에 있어 최소권한의 원칙 위배의 문제점이 해결됨을 보인다.

2. 사용목적 분류를 통한 보안 접근제어 모델

본 절에서는 사용목적 분류를 통한 보안 접근제어 모델로써 Pu-ARBAC을 제시한다. 본 모델은 기존의 ARBAC(Administrative Role Based Access Control) [8]을 기반으로 하고 있으며, 그 중 개인정보에 대한 권한을 그 대상으로 한다.

2.1. Pu-ARBAC 의 환경

사용목적은 개인정보에 대한 권한부여의 단위이며, 본 논문의 접근제어 과정에서는 역할을 통하여 부여된 권한을 이용해 특정 태스크 프로세싱(task processing) 수행 중 필요시 되는 개인의 프라이버시와 관련된 정보를 제공하기 위한 조건이자 권한의 단위

이다. 그림 9는 환경을 예를 들어 설명한다. user는 어느 한 역할에 할당된 후 3가지의 태스크 프로세싱 수행이 가능하며, 이때 개인의 프라이버시와 관련된 정보를 요구하게 되는 상황이다.

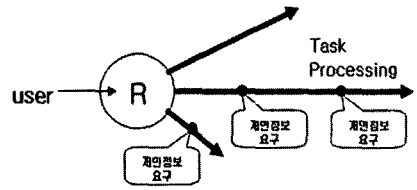


그림 9. Pu-ARBAC 환경의 예

인터넷을 통하여 책을 판매하는 기업이 나이별로 책 선호도를 조사 후 회원들에게 이메일을 사용하여 책을 추천하는 태스크를 수행하는 예를 들면, 여기서 R은 사용자가 얻을 수 있는 역할(role)된다. 또한 역할을 활성화한다는 말은 그에 해당하는 권한을 얻는 것이며, 이때 얻게 되는 권한은 시스템 자원 즉, 나이별로 책 선호도를 분석할 수 있는 분석툴 또는 그 후 고객들에게 책을 소개할 경우 사용되는 이메일 서비스 등을 사용할 수 있는 권한을 말하다. 사용자는 이러한 시스템 자원을 이용하여 일을 수행 중 개인정보를 필요로 하며, 이때 개인정보에 대한 요청시 개인정보에 대한 접근제어가 필요하다. 그림 10의 Pu-ARBAC은 이러한 상황에서 기존보다 향상된 최소권한의 원칙을 따르는 접근제어를 한다.

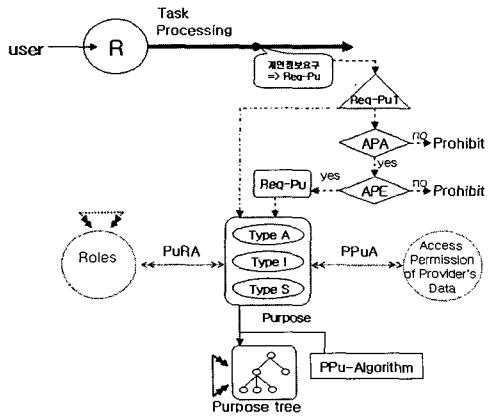


그림 10. Pu-ARBAC 모델

Pu-ARBAC 모델은 사용목적-역할 관리(Purpose-Role Administration)와 권한-사용목적 관리 (Permission-Purpose Administration)의 구성요소로 되어있으며 ARBAC을 따르고 있다. 사용목적 획득과정은 다음과 같다.

- Req-Pu는 태스크 프로세싱상에 관리자의 의해 미리 정해진 사용목적(권한) 이며, Req-Pu↑는 사용목적트리를 참조하여 사용목적 Req-Pu의 상위 사용목적들의 집합을 얻는다.
- APA은 〈ap, cr〉의 2-튜플로 이루어져 있다. cr은 〈r, C〉의 2-튜플을 가지며, C는 역할 속성과 시스템 속성을 말한다. 여기서 APA은 사용자가 요구하는 사용목적 Pu가 활성화 가능하면 yes를 반환하며, 그렇지 않으며 no를 반환한다.
- APE(Access Purpose Existence)은 Req-Pu↑에 Pu이 있으면 yes 그렇지 않으면 no를 반환한다.
- 최종적으로 yes가 반환되면 최초에 정해진 태스크를 진행하는데 필요한 최소의 사용목적(권한) Req-Pu를 선택하여 PPu-Algorithm(사용목적 권한 획득 알고리즘)을 이용하여 권한을 획득하게 된다.

그림 11(a)의 PPu-Algorithm은 [부록B]의 Purpose Language인 **Purpose.xsd** 에 의해서 유효성 검증된 사용목적을 표현한 xml 문서를 이용하여 권한을 획득한다.

그림 11(b)는 사용목적 권한 획득 알고리즘과 시간적 성격의 사용목적에 대한 메타 데이타를 저장하고 있는 Purpose action table 이다.

```
request_purpose(ID)                                   ID : 입력(사용목적ID)
                                                      type : 사용목적 구조
Begin                                                 rights : 권한
  if(search node(ID))
    {
      if (type == "None") then return rights;
      else if (type == "Pu_Link") then go parent's node and return rights;
      else if (type == "Pu_Inheritance_Last") then return rights;
      else if (type == "Pu_Inheritance") then return rights;
      else if (type == "Pu_Precondition_Last") then return rights;
      else if (type == "Pu_Precondition")
        {
          if (comparing condition value with Purpose action table => satisfaction) then return rights;
        }
      else fail;
    }
  else fail;
End
```

(a) PPu-Algorithm(사용목적 권한 획득 알고리즘)

Purpose (id)	수행여부 (yes/no)
Analysis	no
Data Collection	yes
⋮	⋮

(b) Purpose action table

그림 11. 사용목적 권한 획득 알고리즘

Ⅳ. 비교 및 평가

개인의 프라이버시 보호를 다루는 기존의 연구에서는 사용목적에 초점을 맞춰 사용목적에 따라 개인정보에 대한 권한을 부여하였다. 그러나 사용목적에 대한 분류가 미흡하고 그로 인해 최소권한의 원칙이 미흡하게 지켜지고 있다. 본 논문에서는 사용목적 분류를 통한 접근제어 모델을 제시하여 개인의 프라이버시 보호를 위해 최소권한의 원칙을 강화하였다.

제안된 사용목적 분류와 이를 이용한 보안 접근제어 기법을 평가하기 위해서 기존에 제시되었던 모델과 각기 다른 상황의 실험을 통한 비교 분석한다. 그 후에 이들 간에 비교·평가를 통하여 제시한 모델이 좀 더 보완되고, 개선된 점을 보인다.

1. 실험 평가

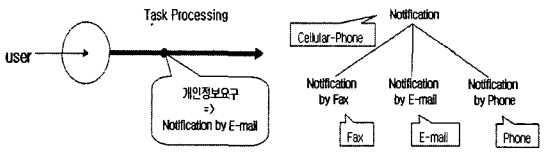
본 실험은 타입 I 접근 방식과 타입 S 접근방식으로 나누어 실험을 한다. 타입 I 접근방식에서는 2.4.2절에서 제시한 문제점에 대해서 기존 모델과 제시한 모델에 대해 실험을 통하여 최소권한의 원칙이 지켜지는지를 비교한다. 타입 S 접근방식에서는 기존 모델에서 제공하고 있지 못하는 특성의 사용목적 구조이다. 이 실험에서는 기존 모델의 메카니즘을 사용했을 경우와 본 논문에서 제시한 모델의 접근 메카니즘을 사용했을 경우 어떠한 차이가 있는지 그리고 최소권한의 원칙을 따르고 있는지를 비교해 본다.

1.1. 타입 I 접근방식

본 절에서는 사용목적기반 접근제어, 히포크라테스 데이타베이스 그리고 Pu-ARBAC 모델간에 타입 I 접근방식에 따른 개인 정보에 대한 프라이버시 보호를 위해 최소권한의 원칙을 준수하는지에 대한 비교분석을 위한 실험을 한다. 이는 세 번째 문제점인 최소권한의 원칙 위배에 대한 실험이다.

실험을 위해 다음 그림 12와 같은 상황을 가정하였다. 사용자는 on-line bookseller에서 회원들에게 E-mail을 통하여 책을 추천하기 위하여 회원들의 E-mail 정보에 대한 권한을 필요로 하는 상황이다. 이때 태스크 프로세스상에는 E-mail에 대한 최소 권한을 가지고 있는 사용목적 Notification by E-mail 이 명시되어 있고, 사용자는 Notification 이하의 모든 사용목적을 활성화할 수 있는 조건을 만족하고 있다고 가정한다.

사용자는 태스크 프로세스상에서 회원들의 E-mail 정보만 가지고 있는 사용목적 Notification by E-mail이 아닌 상위 사용목적인 Notification을 요구하는 경우 그 결과를 각각 모델 별로 실험한다.



(a)개인정보요구상황 (b) 사용목적 트리와 권한

그림 12. 타입 I 접근 방식의 실험을 위한 상황

• 사용목적 기반의 접근제어 모델

사용목적기반 접근제어 모델은 그림 13의 프로세스 과정을 통하여 수행되며, 이때 태스크 프로세스에서 필요로 하는 이상의 권한(Cellular-Phone, Fax, E-mail, Phone)을 부여하게 된다. 이는 최소권한의 원칙을 위배하는 것이며, 개인의 프라이버시를 침해하는 결과를 낳는다.

• 히포크라테스 데이타 베이스 모델

히포크라테스 데이타베이스 모델은 그림 14의 프로세스 과정을 통하여 수행되며, 이때 태스크 프로세스에서 필요로 하는 이상의 권한(Cellular-Phone, Fax, E-mail, Phone)을 부여하게 된다. 이는 최소권한의 원칙을 위배하는 것이며, 개인의 프라이버시를 침해하는 결과를 낳는다.

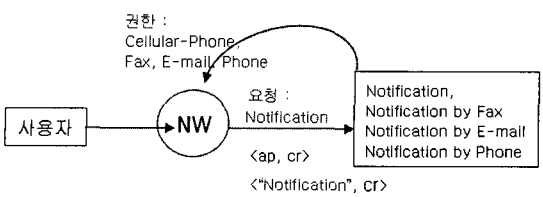


그림 13. 타입 I 접근방식의 사용목적기반의 접근제어

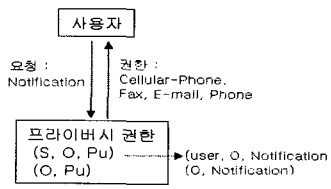


그림 14. 타입 I 접근 방식의 히포크라테스 데이타베이스

• Pu-ARBAC 모델

본 논문에서 제안한 Pu-ARBAC 모델은 그림 15의 프로세스 과정을 통하여 수행되며, 이때 태스크 프로세스에서 필요로 하는 기존의 모델보다 향상된 최소의 권한만을 사용자에게 제공한다. 이는 최소권한의 원칙을 잘 따르고 있으며, 개인의 프라이버시 침해문제에 있어서 이전의 모델과 비교할 때 좀더 개선되고 향상됨을 알 수 있다.

프로세스 과정을 살펴보면, 사용자는 역할 NW을 활성화한 상태이며, 사용목적 Notification을 요구로 시작한다. 사용자는 Notification을 요구하였지만 본 시스템에서는 요구한 사용목적으로 권한을 요구하는 것이 아니라 태스크 프로세스상에서 제시한 사용목적 Notification by E-mail을 사용하여 다음의 과정을 따르며 최종 권한 허용시 Notification by E-mail의 권한이 사용자에게 부여된다.

Notification by E-mail ↑은 사용목적 트리를 참조하여 Notification by E-mail을 포함한 그 상위의 모든 사용목적을 집합으로 갖는다. 즉, Notification by E-mail ↑ = {Notification by E-mail, Notification} 을 말한다.

APA는 요구되어진 사용목적 Notification 이 활성화 가능하면 yes를 그렇지 않으면 no로 결정한다. 가정(사용자는 Notification 이하의 사용목적을 활성화할 수 있는 상황)에 의해 APA 는 yes로 결정된다.

APE는 요구되어진 사용목적 Notification 즉, 활성화된 사용목적인 Notification 이 Notification by E-mail ↑ 에 존재하면 yes를 그렇지 않으면 no 로 결정한다. 여기서는 yes가 된다.

태스크 프로세스상에서 제시한 사용목적 Notification by E-mail이 최종 사용목적으로 결정되며, PPu-Algorithm의 request_purpose(Notification by E-mail)을 통하여 최종 권한을 부여 받는다. 이때 최종적으로 부여되는 권한은 E-mail 이 된다.

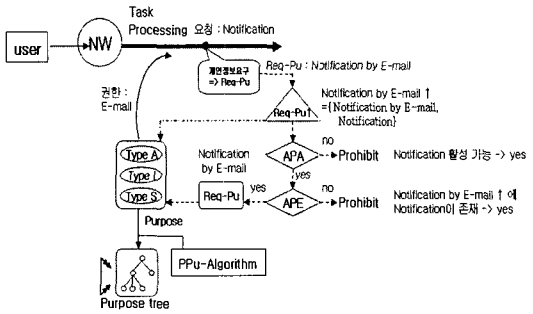


그림 15. 타입 I 접근 방식의 Pu-ARBAC

1.2. 타입 S 접근방식

본 절에서는 사용목적기반 접근제어와 Pu- ARBAC 모델 간에 타입 S 접근방식에 따른 개인정보에 대한 프라이버시 보호를 위해 최소권한의 원칙을 준수하는지에 대한 비교 분석을 하기 위한 실험을 한다. 여기서 히포크라테스 데이타베이스 모델은 사용목적기반 접근제어 모델과 결과가 같기 때문에 생략한다.

실험을 위해 다음 그림 16과 같은 상황을 가정하였다. 사용자는 on-line bookseller에서 신용카드 입금처리에 관련된 일을 하는 직원으로서, 고객의 Transaction_Info, CustomerID 그리고 Credit_Card의 정보를 필요로 하는 상황이다. 이때 태스크 프로세스상에는 사용목적 Transfer by Credit Card이 명시되어 있고, 사용자는 Transfer by Credit Card 이하의 모든 사용목적을 활성화할 수 있는 조건을 만족하고 있다고 가정한다.

사용자는 태스크 프로세스상에서 제시한 사용목적 Transfer by Credit Card를 요구하는 경우 사전조건의 상태에 따라서 결과를 각각 모델 별로 실험한다.

• 사용목적 기반의 접근제어 모델

사용목적기반 접근제어 모델은 그림 17의 프로세스 과정을 통하여 수행되며, 이때 시간적 개념의 부재로 인하여 비즈니스 프로세스상 부여 받아서는 안

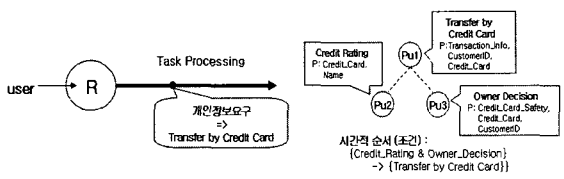


그림 16. 타입 S 접근 방식의 실험을 위한 상황

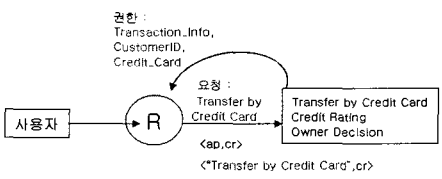


그림 17. 타입 S 접근방식의 사용목적 기반의 접근제어

될 권한(Transaction_Info, CustomerID, Credit_Card)을 부여 받게 된다. 이는 최소권한의 원칙을 위배하는 것이며, 개인의 프라이버시를 침해하는 결과를 낳는다.

• Pu-ARBAC 모델

본 논문에서 제안한 Pu-ARBAC모델은 그림 18의 프로세스 과정을 통하여 수행되며, 이때 비즈니스 프로세스상 부여 받아서는 안될 권한은 부여가 안 된다. 이는 최소권한의 원칙을 잘 따르고 있으며, 개인의 프라이버시를 침해문제에 있어서 이전의 모델과 비교할 때 좀 더 개선되고 향상됨을 알 수 있다.

프로세스과정을 살펴보면, 사용자는 역할 PW을 활성화한 상태이며, 사용목적 Transfer by Credit Card를 요구로 시작한다. 태스크 프로세스 상에서도 사용자가 요구하는 사용목적과 같은 Transfer by Credit Card을 사용하여 다음의 과정을 따르며 최종권한이 허용 시 Transfer by Credit Card의 권한이 사용자에게 부여된다. Transfer by Credit Card ↑ 은 사용목적 트리를 참조하여 Transfer by Credit Card 을 포함한 그 상위의 모든 사용목적을 집합으로 갖는다. 즉, Transfer by Credit Card ↑ = { Transfer by Credit Card } 을 말한다.

APA는 요구되어진 사용목적 Transfer by Credit Card이 활성화 가능하면 yes를 그렇지 않으면 no로 결정한다. 가정(사용자는 Notification 이하의 사용목적을 활성화할 수 있는 상황)에 의해 APA 는 yes 로 결정된다.

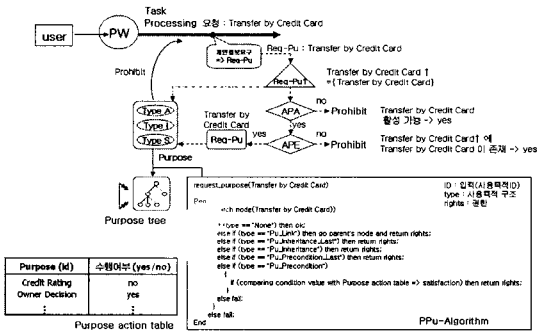


그림 18. 타입 S 접근방식의 Pu-ARBAC

APE는 요구되어진 사용목적 Transfer by Credit Card 즉, 활성화된 사용목적인 Transfer by Credit Card 이 Transfer by Credit Card ↑ 에 존재하면 yes를 그렇지 않으면 no 로 결정한다. 여기서는 yes가된다.

태스크 프로세스상에서 제시한 사용목적 Notification by E-mail 이 최종사용목적으로 결정되며, PPu-Algorithm의 request_purpose (Notification by E-mail)을 이용하여 권한을 요청한다. 사용목적 Notification by E-mail는 타입이 Pu_Precondition이므로 사전조건 {Credit_Rating & Owner_Decision}에 대해 Purpose action table의 값이 각각 yes가 되어야 사용목적 Notification by E-mail 의 권한을 할당해 준다. 그러나 Purpose action table의 Credit Rating 의 값이 no 이므로 최종적으로 사용자는 사용목적 Notification by E-mail 에 대해 권한을 부여 받을 수가 없다.

2. 분석 및 평가

본 절에서는 사용목적의 분류와 이를 이용한 접근제어인 Pu-ARBAC을 기존의 모델과 4.1절의 실험을 통하여 분석한다.

2.4.1절에서 기존 사용목적 관리에 있어서의 문제점인 사용목적에 대한 표현과 사용목적에 대한 삽입·삭제 행위시 사용목적 관리의 미흡함을 3.1.1와 3.1.2절에서 사용목적 분류를 통한 사용목적 관리로 제안한 본 모델이 기존의 문제점을 모두 해결함을 보였다. 이는 기존에 표현 불가능 했던 사용목적을 표현 가능하게 함으로서 개인의 프라이버시와 관련된 정보에 대한 권한을 좀 더 세세히 표현 가능하게 하였다. 또한 관리적 측면에서 Purpose Language와 사용목적 분류를 통한 일관된 정책을 제시하였다.

다음은 기존모델과의 접근제어 관점에서 실험을 통하여 비교 분석한 것이다.

타입 I 에서 비즈니스 프로세스상의 태스크에 필요한 사용목적보다 상위의 사용목적으로 접근시 기존의 모델들은 그 상위 사용목적으로의 접근을 허용하며, 상위 사용목적의 권한을 허가함으로서 최소권한의 원칙을 위배하는 문제점(2.4.2절)을 보였다. 이는 정보제공자의 프라이버시를 해치는 결과를 가져온다. 그러나 제안모델 Pu-ARBAC은 접근은 가능하지만 그 상위 사용목적으로 권한을 부여하지 않고 태스크에서 요구하는 좀 더 강화된 최소의 권한을 제공함으로써 문제점을 해결하였으며, 4.1.1절의 실험을 통하

여 해결됨을 보였다.

또한 타입 S 에서 기존의 모델들은 시간적 개념의 부재로 실제 부여되어서는 안 될 권한을 제공함으로써 개인의 프라이버시를 해치는 결과를 가져왔다. 그러나 제안모델에서는 시간적 개념을 고려함으로써 개인 정보에 대한 권한남용의 문제를 해결하였다.

V. 결 론

최근 들어 사적인 데이타에 대한 개인의 프라이버시 문제가 큰 이슈로 떠오르고 있으며, 그로 인해 개인정보에 대한 수집 방법과 수집 후 보안관리에 대해 많은 연구가 진행 중에 있다. 기존 연구는 개인정보에 대한 프라이버시를 보호하기 위한 방법으로 개인정보에 대한 보안관리로서 권한 단위인 사용목적을 사용한다. 그러나 권한의 단위인 사용목적 사용에 있어서 표현과 관리의 미흡함으로 인하여 실제로 사용자에 대한 접근제어에 있어서 최소권한의 원칙을 따름에 있어 많은 문제점을 가지고 있다.

본 논문은 기존 연구에 대한 문제점을 제시하였고, 또한 문제의 해결책을 제시하였다. 본 논문에서 제안한 기법의 기여도는 다음과 같다. 첫째, 사용목적의 표현력을 향상시키기 위해 사용목적을 3가지 구조로 분류하였으며 둘째로, 각 사용목적 구조에 따른 사용목적의 관리기법을 제시하였다. 셋째로, 이렇게 분류되어 관리되는 사용목적을 통하여 최소권한의 원칙을 따르는 접근제어 기법을 제시하였다.

본 논문에서 제안한 기법은 최근 중요한 이슈로 떠오르는 개인정보에 대한 프라이버시 보안에 있어서 정보 제공자의 의도를 따르며, 보안 관리에 있어서 최소권한의 원칙을 따름으로써 개인정보 제공자의 데이타에 대한 보안을 강화하였다.

본 논문에서 제안한 접근제어 기법은 보통 일반 업무에서 보듯이 태스크상에서 개인정보를 접근하는데 있어 이미 고정화되어 있는 사용목적에 사용자의 업무상 위치나 상황을 조건으로 비교하여 접근 여부를 결정한다. 그러나 상황에 따라서 동적으로 사용목적을 필요로 하는 환경 또한 존재한다. 본 논문은 이러한 환경에 대해 조사가 미흡했고, 또한 그러한 환경에서는 기존의 접근제어 모델들과 차별성이 없다.

본 논문은 한 기업환경에서 개인정보에 대한 프라이버시를 보호하기 위한 기법이다. 그러나 실제로 기업에서 관리하고 있는 개인정보는 한 기업 내에서만 이용되는 것이 아니라 한 기업에서 다른 기업으로 유통되는 경우가 많다. 이러한 경우 개인의 프라이버시 보호에 큰 어려움이 생기며, 이에 대한 제한사항에 대한 연구가 추가적으로 필요하다.

참 고 문 헌

[1] World Wide Web Consortium (W3C). Platform for Privacy Preferences (P3P). Available at www.W3.org/P3P

[2] IBM: Enterprise Privacy Authorization Language (EPAL): Submission request to W3C,http://www.w3.org/Submission/EPAL/, November 2003.

[3] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, Yirong Xu, "Hippocratic Databases", Proceedings of the 28th VLDB Conference, Hong Kong, China, 2002

[4] Ji-Won Byun, Elisa Bertino, Ninghui Li, "Purpose Based Access Control of Complex Data for privacy Protection", SACMAT'05, 2005, Stockholm, Sweden

[5] 이재길, 한욱신, 황규영, "Hippocratic XML Databases: A Model and Access Control Mechanism", 정보과학회논문지: 데이타베이스 제 31 권 제 6호 (2004.12)

[6] Cunter Karjoth, Matthias Schunter, and Michael Waidner, "Privacy-enabled Management of Customer Data", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2004

[7] R.Sandhu, E. Coyne, H. Feinstein, and C. Younman, "Role-Based Access Control Models", IEEE Computer Magazine Vol. 29, 1996, pp.38-47

[8] Ravi Sandhu, Venkata Bhamidipati and Qamar Munawer, "The ARBAC97 Model for Role-Based Administration of Roles", ACM Transactions on Information and System Security, Vol. 2, No. 1, February 1999, Page 105-135

〔부록 A〕 삽입·삭제 함수 알고리즘

```
input_purpose_typeI(Pui){
  while (choose one Puj from Purpose Tree using
    BFS){
    if (Puj ⊂ Pui) then
        input Pui by a parent of Puj and break;
    else if (Puj == Pui)
        then input Pui by a link of Puj and break;
    else if (((Puj ∩ Pui) == some) | ((Puj ∩
      Pui) == none)) then make Purpose Tree as
      typeA and break;
}
}

delete_purpose_typeI(Pui){
 search Pui from Purpose Tree;
 if ((Pui == leaf node)|( Pui == linked node)) then
      delete Pui;
    else if ((Pui ≠ ancestor of Purpose Tree)
      & (Pui ≠ leaf node of Purpose Tree)) then
      delete Pui and arc between parent node of Pui
          and child node of Pui;
    else if (Pui == ancestor of Purpose Tree){
      if (Pui get only one child) then delete Pui;
      else delete Pui and make Purpose Tree as typeA
          with child of Pui;
}
}

Pui : 삽입되어지는 임의의 사용목적,
Puj : 기존 사용목적 트리에 존재하는 임의의 사용목적
```

타입 I - 삽입·삭제 함수 알고리즘

```
input_purpose_typeS(Put){
  while (choose one Puj from Purpose Tree using BFS){
     if (Puj > Put)
         then input Put by a parent of Puj and break;
     else if (Puj == Put)
         then input Put by a sibling node of Puj
           and break;
}

delete_purpose_typeS(Put){
  search Put from Purpose Tree;
  if (Put == ancestor of Purpose Tree){
     if (Put get only one child) then delete Put;
     else delete Put and make Purpose Tree as typeA with
              child of Put;
  }
  else if (Put == leaf node){
if((Put get at least one sibling node) & (condition
     is "or")) then delete Put;
  }
}

Put : 삽입되어지는 임의의 사용목적,
Puj : 기존 사용목적 트리에 존재하는 임의의 사용목적
```

타입 S - 삽입·삭제 함수 알고리즘

〔부록B〕 Purpose Language

```
<?xml version="1.0" encoding="UTF-8" standa
              lone="yes"?>
<xs:schema xmlns:xs="http://www.w3.org/ 2001/
              XMLSchema
  "elementFormDefault="qualified">
  <xs:element name="GeneralPurpose">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Purposes"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Purposes">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Purpose" maxOccurs
          ="unbounded"
           />
        </xs:sequence>
      </xs:complexType>
  </xs:element>
  <xs:element name="Purpose">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Link" minOccurs="0"/>
        <xs:element ref="A_Info"/>
        <xs:element ref="Purpose" minOccurs="0"
              maxOccurs ="unbounded"/>
      </xs:sequence>
      <xs:attribute name="link" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="Yes"/>
            <xs:enumeration value="No"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="type" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="None"/>
            <xs:enumeration value="Pu_Link"/>
            <xs:enumeration value="Pu_
              Inheritance"/>
            <xs:enumeration value= "Pu_
              Inheritance_Last" />
            <xs:enumeration value="Pu_Pre
              condition"/>
            <xs:enumeration value="Pu_Pre
            condition_Last" />
          </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="condition" type="xs:string"
              use="required"/>
    <xs:attribute name="id" type="xs:string"
              use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="Link">
```

```
    <xs:complexType>
       <xs:sequence>
          <xs:element ref="Purpose" maxOccurs=
             "unbounded" />
       </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="A_Info">
    <xs:complexType mixed="true">
       <xs:choice minOccurs="0" maxOccurs=
          "unbounded">
```

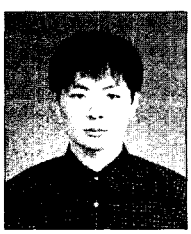
```
          <xs:element ref="X_Path"/>
       </xs:choice>
    </xs:complexType>
  </xs:element>
  <xs:element                    name="X_Path"
              type="xs:string"/>
</xs:schema>
```

Purpose.xsd

〈著者紹介〉

박 석 (Seog Park) 종신회원

1978년 2월 : 서울대학교 계산통계학(이학사)
1980년 2월 : 한국과학기술원 전산학(공학석사)
1983년 8월 : 한국과학기술원 전산학(공학박사)
1983년 9월~현재 : 서강대학교 컴퓨터학과 정교수
1997년 2월~현재 : 한국정보보호학회 이사
1998년 9월~현재 데이터베이스 연구회 운영자문위원
2006년 : 국세청 정보화 자문위원
2005년 : 한국정보과학회 부회장
2004년 1월~2005년 12월 : 한국정보과학회지 편집위원장
2002년 ~ 2004년 : University of Virginia 방문교수
2006년 : VLDB Panel Co-chair
2006년 : KCC 2006 한국컴퓨터종합학술대회 프로그램위원장
1999년 ~ 현재 : DASFAA Steering Committee
2004년 : DASFAA 2004 Organization Chair
〈관심분야〉 데이터베이스 보안, 트랜잭션 관리, 센서네트워크 데이터 관리, XML, 스트리링 데이터 처리, 유비쿼터스 컴퓨팅, 역할기반 접근제어, 상황-인식 접근제어



나 석 현 (Seok-Hyun Na) 학생회원

2004년 2월 : 국민대학교 컴퓨터학과(학사)
2006년 2월 : 서강대학교 컴퓨터학과(석사)
〈관심분야〉 역할기반 접근제어, 상황-인식 접근제어, 데이터베이스 보안, 웹 서비스 보안, XML