논문 2010-6-30

불확정 상황정보 상에서의 접근제어 방식

A Method for Access Control on Uncertain Context

강우준*

Woo-Jun Kang

요 약 새로운 정보기술의 발전으로 인해 정보 접근과 획득 방식이 훨씬 다양하고 용이해지고 있는 반면 다양하고 성능 좋은 도구를 이용한 불법적인 접근이 가능하도록 하는 부작용이 초래되고 있다. 이러한 위협에 대응하는 데이터베이스 기술로는 접근제어가 있고 현재 새로운 컴퓨팅 환경에 대응하기 위해 전통적인 접근제어를 확장한 다양한 연구들이 수행되고 있다. 본 연구에서는 상황정보의 시멘틱 정보를 기반으로 보안정책에 의해 명시된 상황제약조건이 질의에 수반되는 상황제약조건의 구문과 일치하지 않는 경우에도 적절한 보안정책 집행이 가능하도록 하는 접근제어 방식을 제안한다. 상황에 대한 의미적 정책집행을 위해 시멘틱 트리계층구조 상에서 이들 간의 의미적 함의관계를 이용하고 함의관계에 의해 초래될 수 있는 과도한 권한부여를 방지하기 위해 의미 차를 정량적으로 측정할 수 있는 인수를 정의하여 설정된 시스템 정의 임계치 범위 내에서만 의미적 함의에 의한 권한부여가 이루어지도록 한다.

Abstract New information technologies make it easy to access and acquire information in various ways. However, It also enable powerful and various threat to system security. The prominent database technology challenging these threats is access control. Currently, to keep pace with the new paradigms, new extended access control methods are challenged. We study access control with uncertain context. With respect to access control, it is possible that there is a discrepancy between the syntactic phrase in security policies and that in queries, called semantic gap problem. In our semantic access control, we extract semantic implications from context tree and introduce the measure factor to calculate the degree of the discrepancy, which is used to control the exceed privileges.

Key Words : 접근제어, 상황인지, 온톨로지.

Ⅰ. 서 론

오늘날 정보기술의 눈부신 발전은 그 어느 때 보다 원활하고 자유스럽게 정보의 배포와 공유가 가능하도록 하고 있다. 언제 어디서나 정보를 획득할 수 있도록 하는 유비쿼터스 기술[1], 상황정보를 기반으로 하여 보다 양질의 서비스를 제공하기 위한 상황인지 기술[2], 기계로 하여금 정보의 의미를 파악할 수 있도록 하는 시멘틱[3] 기술 등 새로운 정보기술 패러다임들과 다양한 기술들이 지속적으로 제안, 개발되고 있고, 이러한 기술 발전에 따라 사용자의 정보 접근과 획득 방식이 이전보다 다양하고 용이해지고 있다. 하지만 그 이면에는 불법적인 접근 주체로 하여금 이전보다 더 나은 성능의 도구를 사용하여 더 다양한 방식으로 침입할 수 있도록 하는 부작용을 낳고 있으며, 이에 따라 시스템의 안전과 개인의 사적 자유영역에 심각한 위협을 가하고 있다. 이러한 보안 위협에 대응하기 위한 기술 중 데이터베이스를 위한 대표적인 보안기술로 접근제어(Access Control)가 있다. 현재 새로운 컴퓨팅 환경에 대응하기 위해 전통적 접근제어를 확장한 상황인지 접근제어, 프라이버시보호 접근제어, 그리고 XML 접근제어 등의 다양한 연구들이 수행되고 있다.

*정회원, 그리스도대학교 경영학부 경영정보학전공
접수일자 : 2010.10.25, 수정완료일자 : 2010.11.25
게재확정일자 2010.12.15

데이터베이스 보안을 위해서는 특정 접근주체가 어떤 접근대상에 대해 어떠한 권한을 행사할 수 있는지를 명시하고 이를 집행하는 접근제어 기법이 요구된다. 전통적인 방식에서는 보안정책 상에 명시되는 정보와 집행에 필요한 정보들에 불확실성이 존재하지 않아 명확하게 정책집행을 수행할 수 있었다. 하지만 유비쿼터스나 클라우딩 컴퓨팅과 같은 새로운 패러다임이 등장함에 따라 이러한 정보들에 불확실성이 존재하게 되었고, 불확실성이 내재된 정보에 대한 접근제어 방식이 필요하게 되었다.

정책집행에 있어 필요 정보인 문맥 정보에 내재된 불확실성을 반영 또는 제거하는 방법에 대한 연구와 이러한 불확실한 정보의 확실성 정도에 따라 차별화된 데이터 공개방식에 대한 연구와 이전의 접근제어 기법과 통합할 수 있도록 하는 연구가 필요하다. 외국의 데이터베이스 벤더인 Oracle, IBM, Microsoft 사 등의 연구소에서 이론적인 기반을 다지기 위한 시도들을 하고 있다[4,5,6]. 국내에서는 접근제어 기법으로 역할기반 접근제어에 대한 연구는 활발히 이루어지고 있으나 불확실성을 처리할 수 있는 데이터베이스 보안에 대한 연구는 극히 미비한 수준이다.

본 논문에서는 불확정 상황인지와 접근제어를 통합한 모델을 제시한다. 제안 모델에서는 상황의 개념적 의미를 기반으로 의미적 정책집행을 수행한다. 여기서 의미적 정책집행이란 보안정책에 의해 명시된 상황제약조건의 구문이 질의에 수반되는 상황제약조건의 구문과 일치하지 않는 경우에도 적절한 정책집행이 가능하도록 한다는 것을 의미한다. 상황에 대한 의미적 정책집행을 위해서 상황 간의 관계를 표현하는 온톨로지로부터 추출된 트리 계층구조 상에서 이들 간의 의미적 함의관계를 이용하는 방식과 의미적 함의에 의해 초래될 수 있는 과도한 권한부여를 제한하기 위한 방법을 제시한다. 과도한 권한부여를 제어하기 위해 상황 개념 간의 의미 차를 정량적으로 측정할 수 있는 인수를 정의하고 이 인수를 기반으로 미리 정의된 시스템 정의 임계치 범위 내에서만 함의에 의한 권한부여가 가능하도록 한다.

본 논문에서의 제안 방식은 불확실성이 본질적으로 내재될 수 있는 모바일, 유비쿼터스, 클라우딩 그리고 상황인지 환경에서 데이터베이스 보안에 적용될 수 있으리라 사료된다.

본 논문의 구성은 다음과 같다. 2장에서는 배경지식이 되는 접근제어, 상황인지와 관련된 연구들을 소개한다. 3장에서는 제안 모델의 정의와 계층구조, 의미적 상황인지를 위한 개념과 이러한 개념들을 바탕으로 하는 정책집행 알고리즘에 대해 설명하고 4장에서 결론을 맺는다.

Ⅱ. 배경지식

1. 접근제어

접근제어란 컴퓨터시스템 또는 네트워크 상의 자원에 대한 접근을 제어함을 의미한다. 전통적 접근제어 메커니즘은 크게 세 종류 즉, 강제적 접근제어, 임의적 접근제어 그리고 역할기반 접근제어[7]로 구분할 수 있다.

강제적 접근제어 방식은 군사적 응용의 보안에 적합하며, 데이터와 사용자에 대해 보안등급이 부여 되는데 데이터의 보안등급은 등급, 사용자의 보안등급은 자격이라 한다. 이 방식의 보안 정책에서는 특정 등급의 데이터에 대한 접근을 인가 받기 위해서는 주체가 어떤 자격 이상이 되어야 하는지를 명시하게 된다. 명시된 보안정책과 Bell-Lapadular 모델에서 제안하는 단순속성과 스타속성이라는 제약조건에 의해 접근의 허가가 결정된다. 이 방식은 보안 관리자로 하여금 허용되는 연산들을 명시적으로 정하게 함으로써 보안수준을 향상시킬 수 있다.

임의적 접근제어 방식은 접근허가를 특정 주체에게 선별적으로 부여하고 철회할 수 있도록 한다. 또한 직접적으로 권한 허가를 받은 사용자가 데이터 제공자의 동의 하에 또 다른 사용자에게 권한을 양도할 수 있도록 한다. 대부분의 상용 시스템에서는 접근제어 리스트를 이용하여 접근제어를 관리하는데, 대규모 사용자가 접속하는 시스템에서는 각 사용자 별로 접근허가를 관리해야 하므로 매우 비효율적이다. 또한 조직 내 사용자의 역할이 변경되었을 경우 전체적으로 시스템 보안설정을 변경해야 하는 문제가 발생하기도 한다.

전통적인 강제적 접근제어와 임의적 접근제어에서 발생하는 정책관리의 문제점을 해결하기 위한 대안으로 널리 인정되고 있는 방식이 역할기반 접근제어 방식이다. 역할기반 접근제어를 위해 소개된 다양한 방식들 중 현재 널리 사용되고 있는 방식은 미 상무부 산하 표준기관인 NIST에서 제정한 역할기반 접근제어[7]이다. 이는 각 주체에게 권한을 직접 부여하기 보다는 주체를 조직 구

성을 반영하는 역할에 연관시킨 후 역할에 권한을 부여함으로써 정책관리를 단순화하고 구현 시 유연한 확장성을 제공한다. 이외에도 의무분리와 최소권한 제약을 지원하여 권한의 오용이나 남용을 방지할 수 있도록 하고 정책 간 충돌을 해결하기 위한 금지우선과 허용우선이라는 두 가지 정책충돌해결 방식을 지원한다. 특히, 역할 계층구조를 기반으로 상위 역할이 하위 역할의 권한을 상속 받도록 하여 새로운 정책을 함의(유도) 할 수 있도록 한다. 또한 유도되는 암시적 정책이 명시적 정책과 충돌하는 경우를 위해 전파충돌 해결기법을 지원하고 있다.

2. 상황인지

유비쿼터스 컴퓨팅 그룹에서는 상황을 환경과 사용자 간의 상호작용에 관련된 모든 환경, 주체, 객체 그리고 조건에 대한 정보라고 정의하고 있다. 상황의 손쉬운 예제는 장소, 시간, 온도 등이 될 수 있으며, 상황인지 인프라에 요구되는 필수 기능으로는 상황정보의 획득과 전달 기능, 복잡한 추론을 지원하기 위한 강력한 형식적 상황 모델과 다양한 상황 환경에 응용이 손쉽게 적용될 수 있도록 인터페이스를 제공해야 하는 기능들이 있다. 상황 미들웨어 GAIA 상에서 개발한 상황인지 시스템[8]은 상황과 상황변화를 일차서술논리를 기반으로 기술하고 처리한다. 일차서술논리는 논리곱, 논리합, 부정 그리고 전체 한정자와 존재 한정자 등의 연산을 지원하며 센서로부터 획득되는 기본 상황에 연역 추론을 적용하여 상위 개념의 상황을 유도 할 수 있도록 지원한다. 하지만 이러한 장점에도 불구하고 구현에 있어서 불완전하고 비결정적인 약점을 지니고 있기 때문에 실제로는 그것의 부분집합인 기술논리(Description Logic)를 주로 사용한다.

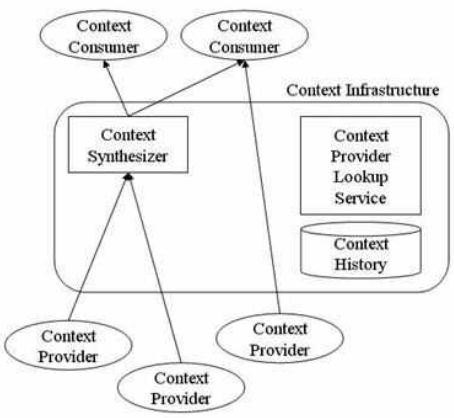

그림 1. GAIA 상황인지 미들웨어
Fig. 1. GAIA Context-aware Middleware

그림 1은 상황인지 미들웨어인 GAIA[8]의 구조를 보여주고 있다. 상황인지 미들웨어는 상황의 획득, 합성 그리고 분배를 담당한다. 센서 기기로부터 기본 상황을 수집하고 수집된 기본 상황으로부터 응용이 필요로 하는 상위 개념의 상황으로 합성한다. 그 후 앞서 언급한 상황 전달 프로토콜을 이용하여 상황을 전달한다. 상황 사용자는 자신이 필요로 하는 상황 항목을 디렉토리 서비스의 일종인 룩업 서비스를 이용하여 검색한다.

Ⅲ. 불확정 상황인지 접근제어

이 장에서는 본 연구에서 제안하는 불확정 상황정보를 접근제어의 정책집행에 반영할 수 있도록 하는 방식에 대해 설명한다.

1. 상황 개념 계층구조

계층구조는 개념과 개념 간의 관계를 표준형식으로 기술하는 온톨로지로부터 추출될 수도 있고 보안관리자나 응용관리자가 명시적으로 구성할 수도 있다. 온톨로지로부터 추출될 수 있는 추론규칙들은 Qin 등에 의해 제안되었으며 그림 2는 온톨로지로부터 추출될 수 있는 추론규칙의 종류를 보여주고 있다. 그리고 그림 3은 병원의 병실배치 상황에 관한 온톨로지를, 그림 4는 해당 온톨로지에 추론규칙을 적용하여 생성되는 상황계층트리를 보여주고 있다.

Relationship between concepts	Inference rule
EQUIVALENCE	if $C_i \equiv C_j$ then $C_i \Rightarrow C_j$
IS-PART-OF	if $C_j \in \{C_i\}$ then $C_i \Rightarrow C_j$
IS-A	if $C_i \subset C_j$ then $C_i \Rightarrow C_j$
UNION	if $C_j = C_1 \cup C_2 \cup ... \cup C_k$ then $C_i \Rightarrow C_j$, i=1,...,k
INTERSECTION	if $C_i = C_1 \cap C_2 \cap ... \cap C_k$ then $C_i \Rightarrow C_j$, j=1,...,k

그림 2. 온톨로지로부터 유도되는 추론규칙
Fig. 2. Inference Rules derived from ontologies

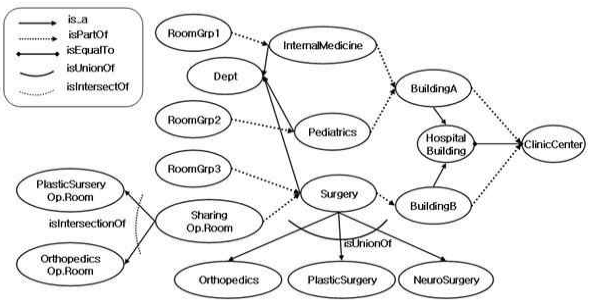

그림 3. 병실 배치 상황에 관한 온톨로지 예
Fig. 3. Example of ontology about hospital placements

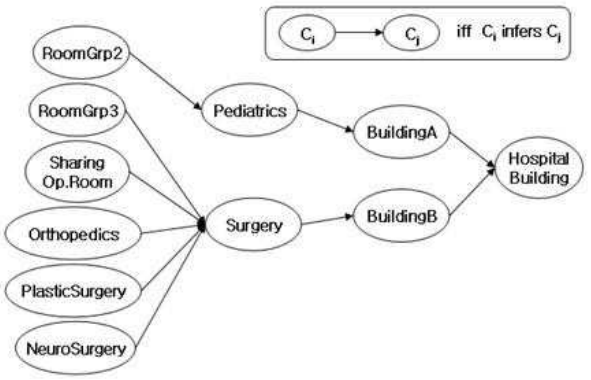

그림 4. 병실 배치에 대한 상황계층트리
Fig. 4. Context hierarchy tree about hospital placements

2. 의미 차 측정

계층트리 상의 두 개념 간의 의미 차를 정량적으로 측정하기 위해 측정인수 SG를 정의한다. SG는 시스템 정의 임계값을 설정하는 기준이 되며 함의를 제한함으로써 보안요구사항인 기밀성과 가용성의 상충관계를 관리할 수 있도록 한다.

엔트로피 E를 기반으로 하고, 비율 계산을 위해 변형한 형태 $2^{E(C)}$를 사용한다. C는 계층트리 상의 인스턴스(단말노드) 또는 타입(비단말노드)이고, E(C)는 C의 엔트로피이다. 만약 C가 인스턴스이면 E(C) = 0 이고 따라서 $2^{E(C)} = 1$ 이 된다. 만약 C 가 타입이면 $2^{E(C)}$ 는 해당 타입 C에 특정 인스턴스가 속하는지를 검사하기 위한 계산 비용을 의미하게 된다. 여기서 특정 인스턴스는 타입 C가 속하는 경로의 단말노드이다. 계층트리에서 같은 경로에 속한다는 것은 의미적으로 관계가 있다는 것을 뜻한다.

정의 1. (의미적 연관) 트리구조에서 경로는 트리의 루트노트에서 단말노드까지의 연결에 포함되는 모든 노드들의 집합으로 정의된다. '계층트리 HT 상에서, 만약 두 노드 Ci, Cj가 동일한 경로에 속하면, C_i, C_j는 의미적으로 연관되었다' 라고 한다.

정의 2. (의미 차) '계층트리 HT 상에서, 만약 C_i, C_j가 의미적으로 연관되었고 i≠j 이면 Ci, Cj간에는 의미 차(Semantic Gap)가 존재한다' 라고 한다.

정의 3. (의미 차 측정 인수) 인수 SG 는 다음과 같이 정의된다.

$SG(C_{i,}C_j) = \frac{2^{E(C_i)}}{2^{E(C_j)}}$, 여기서 C_i, C_j는 계층트리 HT 상에서 의미적으로 연관되었고 C_i는 C_j의 조상이므로 SG는 항상 1보다 크거나 같다(SG≥1).

예제 1. 상황트리계층 CHT 상에서 서로 연관되어 있는 두 상황타입 외과병동과 3호그룹병실이 각각 20개와 5개의 인스턴스(여기서는 병실)을 포함하고 있다면,

$$E(\text{외과병동}) = -\sum_{x \in \text{외과병동}} p(x)\log_2 p(x)$$

$$= 20 \times (\frac{\log_2 20}{20}) = \log_2 20$$

따라서 $2^{E(\text{외과병동})} = 20$ 이 되며, 3호그룹병실에 대해서도 동일하게 계산하면 $2^{E(\text{3호그룹병실})} = 5$ 가 된다. 그러므로 외과병동과 3호그룹병실이라는 두 개념 간의 의미 차는 *SG(외과병동, 3호그룹병실) = 4* 가 된다.

정리 1. 인스턴스들이 타입에 속할 확률이 균등분포를 따른다면 각 타입에 속하는 인스턴스의 갯수 즉, 도메인 크기의 비가 SG가 된다.

$SG(C_i, C_j) = \frac{|C_i|}{|C_j|}$, 여기서 |Cj|≠0 이고 |C|는 타입 C의 도메인 크기를 나타낸다.

증명. 임의 n개의 인스턴스가 동일한 확률로 타입 Ci에 속한다고하면 하나의 인스턴스가 타입 Ci에 속할 확

률은 1/n 즉 1/|Ci| 이다. 마찬가지로 임의 m개의 인스턴스가 동일한 확률로 타입 Cj 에 속한다고하면 하나의 인스턴스가 타입 Cj에 속할 확률은 1/m 즉, 1/|Cj| 이다. 따라서,

$$E(C_i) = \sum_{x \in C_i} \frac{1}{|C_i|} log_2 |C_i|$$

$= |C_i| \times (\frac{1}{|C_i|} log_2 |C_i|) = \log_2 |C_i|$ 이다. 따라서 $2^{E(C_i)} = |C_i|$ 이다. 위와 같이 적용하면 $2^{E(C_j)} = |C_j|$ 이다. 그러므로 $SG(C_i, C_j) = \frac{2^{E(C_i)}}{2^{E(C_j)}} = \frac{|C_i|}{|C_j|}$ □

예제 2. 앞의 예제 1에서 추가적으로 '각 인스턴스가 의미적으로 관련된 타입에 포함될 확률이 균등분포을 따른다' 라는 조건이 주어지면, *SG(외과병동, 3호그룹병실)* 를두 타입의 도메인 크기의 비 */외과병동//3호그룹병실/ = 4* 로 간단히 계산 할 수 있다.

3. 의미적 상황인지 접근제어

접근제어 시스템에서 의미적 상황인지가 필요한 경우를 유비쿼터스 병원의 예를 들어 살펴보자. 접근제어 정책에 정의된 접근규칙이 '의사 갑이 외과병동에 있을 때에만 환자 병에 대한 의료기록에 대해 접근할 수 있다' 이고 접근질의는 '의사 갑이 환자 병의 의료기록에 대한 조회를 요청한다' 이고 접근질의에 수반된 상황정보는 '의사 갑의 현재 위치는 3호병실그룹' 이라고 할 때, 기존 접근제어에서는 정책집행 시 접근규칙에 명시된 상황 제약사항인 '외과병동' 과 질의에 수반된 상황 '3호병실그룹' 를 구문 수준에서만 비교한다. 이 경우에는 양쪽 상황의 구문이 일치하지 않으므로 접근 요청을 거절한다. 하지만 3호그룹병실이 외과병동 내에 위치하고 있다는 관계를 가지고 있다면 접근을 허용하는 것이 타당하다. 본 연구에서는 이런 상황 개념 간의 의미적 관계를 고려한 접근제어가 집행될 수 있도록 하는 의미적 상황 제약조건 평가 방법을 제안한다.

상황인지 접근제어 시스템에서는 접근요청 시 접근제어 모델의 기본 요소인 S, O, A에 대한 권한을 검사하고 이 검사가 통과되면 제약조건으로서 상황조건을 검사하게 된다. 즉, 접근질의가 <S′, O′, A′, C′> 라면, 상황인지 접근제어 시스템은 우선 기본접근검사를 수행하여 <S′, O′, A′> 와 부합되는 <S, O, A>가 보안정책에 명시되어 있는지를 검사한다. 만약 구문적으로 정확하게 일치하는 정책규칙이 있으면 상황제약에 대한 검사를 다시 수행하게 되고 없으면 접근을 거부하게 된다. 앞서 1절에서 살펴보았던 개념계층트리는 정책과 질의에 포함되는 기본 요소들의 개념이 구문적으로 정확하게 일치하지 않아도 접근제어가 가능하도록 하여 보안정책의 명세, 유지, 관리를 용이하게 한다. 역할기반 접근제어에서 주체 S′ 가 계층구조 상에서 S 의 상위역할이라면 접근을 허가하는 것이 그 대표적인 예이다. 이러한 계층구조를 O나 A에 대해서도 확장할 수 있다. 하지만 본 논문에서는 상황에 대한 계층구조만을 고려한다.

기존 모델에서의 상황제약검사는 정책에 기술되어 있는 상황과 질의에 수반되어 있는 상황이 일치하는지를 검사하고 두 상황이 구문적으로 일치하면 최종적으로 접근을 허가하게 된다. 하지만 이 방식은 두 상황이 의미적으로 상위, 하위 개념으로 연결되어 있다고 해도 이를 고려하지 않고 있다. 이를 해결하기 위하여 상황에 대한 계층구조를 생성하고 계층구조 상에서 상황개념 간의 관계를 유추할 수 있도록 하는 방식을 제안한다. 다음은 제안하는 방식을 위한 정의들이다. 3장 1절에서 보았던 상황계층트리와 트리 상에서 선조, 후손 그리고 혈통의 관계를 다음과 같이 정의한다.

정의 4. (상황계층트리) 상황이란 응용시스템과 사용자 간의 상호작용에 관련된 모든 환경, 주체, 객체 그리고 조건에 대한 정보이다. 상황집합 C는 트리 형태의 계층구조로 조직되고 이를 트리 형태의 상황계층구조 줄여서, 상황계층트리 CHT라 한다. CHT 상의 각 노드는 C 상의 상황들을 나타내며 간선은 두 상황 간의 상위와 하위개념의 순서를 나타낸다. C_i, C_j 를 CHT 상의 목적이라 할 때, '만약 CHT 에 C_i 에서 C_j 로의 하향경로가 존재하면 Ci는 Cj의 조상이다 또는 C_j 는 C_i 의 후손이다' 라고 한다.

정의 5. (선조, 후손, 혈통) 상황계층트리에서 선조, 후손, 혈통(lineage)은 다음과 같이 정의된다. CHT를 상황계층구조, CS 를 CHT 상에 있는 상황들의 집합이라고 할 때, 임의의 상황집합 C⊆CS 에 대해, ANC(C)는 C에 속하는 각 노드들의 조상노드들을 합한 집합이며 해당 집합에는 C 자신도 포함된다. DSC(C)는 C에 속하는 각 노드들의 후손노드들을 합한 집합이며 해당 집합에는 C

자신도 포함된다. LNG(C)는 C에 속하는 각 노드들의 조상과 후손을 모두 합한 집합이다. 즉 LNG(C) = ANC(C) ∪ DSC(C) 이다.

직관적으로, 데이터에 대한 접근질의 시 해당 데이터에 대해서 보안정책 상에 명시된 상황이 질의에 수반된 상황을 명시적으로 포함하거나 암시적으로 포함하게 되면 해당 데이터에 대한 접근요청이 허용되어야 함을 알 수 있다. 즉 정책상황은 보안정책에 의해 명시적으로 포함된 상황뿐만 아니라 상황계층구조 상에서 암시적으로 내포하게 되는 상황들까지도 모두 포함해야 한다.

정책상황의 정의에 있어 또, 한 가지 고려할 점은 유연한 정책의 기술과 용이한 정책 관리를 위해서 금지규칙을 지원해야 한다는 것이다. 하지만 금지규칙을 허용하게 되면 허용규칙과의 정책충돌이라는 문제가 발생하게 된다. 제안하는 방식에서는 정책상황에 대해 금지규칙과 허용규칙을 모두 지원할 수 있도록 금지정책상황(Negative Policy-specific Context NPC) 과 허용정책상황 (Positive Policy-specific Context PPC) 을 정의한다. 금지정책상황과 허용정책상황 간에 정책충돌 시에는 거부우선 정책을 적용함으로써 해결한다. 즉 상황 간 정책충돌 시에는 금지정책상황 NPC가 허용정책상황 PPC 보다 우선한다.

계층트리에서는 부모노드가 자식노드의 개념을 포함한다. 따라서 특정 노드에 허용규칙을 적용하면 하향으로 함의를 수행한다. 이렇게 허용규칙을 적용할 때의 함의는 간단하지만 금지규칙이 적용할 때의 함의는 두 가지 의미를 갖는다. 첫째로, 허용규칙을 적용할 때와 같이 상위개념의 상황에서 접근이 금지된다면 이 개념에 포함되는 하위개념에서도 접근이 금지되어야 한다는 의미를 갖는다. 둘째로, 형식논리에서의 대우절의 의미를 적용하면 하위개념의 상황에서 접근이 금지된다면 상위개념에서도 상황에서도 금지되어야 한다. 따라서 계층구조 상의 한 개념에 금지규칙을 적용할 때에는 상향과 하향, 양방향으로 함의를 수행하여야 한다.

본 연구에서는 기존의 Byun[10]에서 제안한 기본개념을 바탕으로 상황의 의미를 반영하는 효율적인 정책집행이 가능하도록 하는 방법을 제안한다.

정의 6. (질의상황, 정책상황, 함의정책상황) 보안정책의 규칙에 기술되어 있는 상황을 정책상황(Policy-specific Context PC)이라 하고 접근질의 시 질의에 수반되어 있는 상황을 질의상황(Query-specific Context QC)라 한다. CHT 를 상황계층트리, C 를 CHT 상의 상황개념들의 집합이라고 할 때, 특정 접근대상 D 에 대한 정책상황 PC는 2-튜플 <PPC, NPC> 로 정의된다. 여기서, PPC⊆C 는 허용규칙이 적용되는 상황들의 집합이며 NPC⊆C 는 금지규칙이 적용되는 상황들의 집합이다. 특정 접근대상 D 에 대한 함의를 포함한 모든 정책상황의 집합을 함의정책상황 PC* = DSC(PPC) - LNG(NPC) 라 한다.

예제 3. 그림 5의 CHT 에서 PC=<{Building B}, {Sharing Op. Room}> 라고 하면, DSC(PPC) = DSC(Building B) = { Building B, Surgery, RoomGrp3, Orthopedics, Sharing Op. Room, Room301, ..., Room305, RoomS01, ..., RoomS05, Room105, ..., Room110 } 이고, LNG(NPC) = LNG(Sharing Op. Room) = {Hospital Building, Building B, Surgery, Sharing Op. Room, Room105, ..., Room110 } 이다. 따라서 PC* = { RoomGrp3, Orthopedics, Room301, ..., Room305, RoomS01, ..., Rooms05 } 이다.

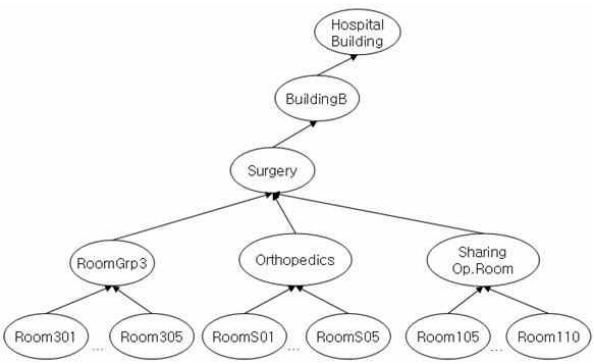

그림 5. 인스턴스가 포함된 상황계층트리
Fig. 5. Context hierarchy tree with instances

기본접근검사 후 수행되는 상황제약검사의 결과는 정책상황 PC와 질의상황 QC간의 관계에 의해 결정된다. 질의상황 QC가 함의정책상황 PC* 에 포함되면 접근이 허용되며 이 경우 '질의상황이 정책상황을 준수한다'라고 한다.

정의 7. (정책상황 준수) 상황계층트리 CHT 상에서 정책상황이 허용정책상황과 금지정책상황의 쌍 PC

<PPC, NPC>로 정의되고, 접근질의에 수반된 질의상황이 QC 라고 할 때, 만약 QC ∈ PC* 이면 질의상황 QC는 정책상황 PC를 상황계층트리 CHT 상에서 준수한다.

예제 4. 그림 3의 CHT에서 PC가 <{Building B}, {Sharing Op. Room}> 이고QC가 Surgery이면, PC는 QC를 준수하지 않는다. 왜냐하면 PC* = {RoomGrp3, Orthopedics, Room301, ... , Room305, RoomS01, ... , Rooms05} 이고 QC ∉ PC* 이기 때문이다. 만약 QC가 RoomGrp3 이라면 QC ∈ PC* 를 만족하므로 QC는 PC를 CHT 상에서 준수한다.

기밀성이라는 보안 요구사항을 고려할 때, 계층트리 상에서 수행되는 함의는 과도한 권한부여라는 문제점을 초래한다. 이 문제를 해결하기 위해서는 함의의 정도를 제한하는 것이 필요하다. 금지규칙의 함의는 그 자체가 권한을 제한하고 있으므로 제한의 대상이 되지 않지만 허용규칙의 함의는 해당 상황개념의 모든 후손노드에게 권한이 부여되므로 제한의 대상이 된다. 계층트리 상에서 함의를 제한하는 가장 단순한 방법은 함의의 정도를 트리의 레벨로 표현하는 방식이다. 즉 어떤 개념의 함의를 구하고자 할 때 해당 개념이 속해 있는 트리 레벨에서 하향으로 어느 정도 레벨까지 함의를 허용할 것인가를 정의하는 것이다. 예를 들어, 그림 5의 계층트리에서 개념 'Building B' 에 대한 정책상황의 함의를 구한다고 하자. 이때 시스템 수준에서 함의 가능 정도를 1이라고 정했다면 ,트리 상에서 다음 레벨의 'Surgery' 까지만 함의에 포함하고 그 하위의 개념들은 포함하지 않는 방식이다. 이 방식은 단순하지만 각 노드 간의 의미를 정량적으로 표현하지 못하므로 정확한 제어가 불가능하다. 제안 방식에서는 정량적 기준으로 함의를 제한하기 위하여 앞서 소개한 SG를 이용하여 상황 간의 의미 차를 정량적으로 측정하고 이를 기반으로 시스템 정의 임계치를 설정하여 한계치 범위 내에서만 함의를 허용함으로써 권한부여가 정량적으로 제한 되도록 한다. 그림 6은 후손노드 중 제한함의를 만족하는 노드로 구성되는 집합인 제한후손(Limited Descendent LDSC)을 구하는 알고리즘을 보여주고 있다.

정의 3.13. (제한함축정책상황) 함축정책상황에 함축제한이 적용될 때 이를 제한함축정책상황 LPC^* = LDSC(PPC) – LNG(NPC) 라 한다.

정의 8. (제한정책상황 준수) 상황계층 CHT 를 기반으로 정책상황 PC <PPC, NPC>가 명시되고, 시스템 정의 제한함축 임계치 TH 가 설정되었고, 접근질의에 수반된 질의상황이 QC 라고 할 때, '만약 QC ∈ LPC^* 이면 질의상황 QC는 정책상황 PC를 상황계층트리 CHT 상에서 함축 제한을 만족하면서 준수한다' 라고 한다.

이러한 상황 간 의미차를 반영하는 의미적 상황제약 평가 알고리즘을 그림 7에 제시하고 있다.

```
Algorithm (Limited Descendent)
LDSC(C, CHT, TH)
IN:   a set C of nodes ⊆ CHT,
      concept hierarchy tree CHT,
      system-defined threshold TH
OUT:  a set LDSC of nodes ⊆ CHT
{
      LDSC = {};
      FOR each c ∈ C {
            i = the level of node c over CHT;

            // Ni, located in the same path of c, is a node at level i of CHT
            WHILE( SGD(c, Ni ) < TH ) {
                  LDSC = LDSC ∪ Ni ;
                  i = i+1;
            }
      }
      RETURN( LDSC );
}
```

그림 6. 제한후손 생성 알고리즘
Fig. 6. Algorithm for generation of LDSC

```
Algorithm (Semantic Context Constraints Evaluation)
SCCE (QC, PC)
IN:   a set of query-specific contexts QC, a set of privacy-specific context PC
      PC is 2-tuple <PPC, NPC>, where PPC, NPC denote Positive PC,
      Negative PC, respectively.
OUT:  PERMIT / DENY
Description: context constraints is evaluated for the O specified
            by access query <S, O, A, QC, P>
{
      IF( QC ∈ (PC* = LDSC(PPC) - LNG(NPC)) ) {
            RETURN (PERMIT);
      }
      ELSE {
            RETURN (DENY);
      }
}
```

그림 7. 의미적 상황제약 평가 알고리즘
Fig. 7. Algorithm SCCE for evaluation of semantic context constraints

의미적 상황을 고려한 접근제어에서의 정책집행을 위한 최종적인 알고리즘은 그림 8과 같다. 첫 번째로는 기본접근검사 함수를 실행하여 질의에 포함된 <S′, O′, P′>가 보안정책 데이터베이스에 저장되어 있는 해당 접근제어 규칙 <S, O, P>와 부합되는지를 검사한다. 검사가 통과되면 두 번째로 의미적 상황제약조건 평가 알고리즘인 SCCE 를 수행하여 질의상황 <C′>가 정책데이터베이스에 저장되어 있는 정책상황 <C>를 시스템에서 정의한 임계치 범위 내에서 준수하는지 검사한다. 결국, 접근제어 정책집행에 있어 이 두가지 검사를 모두 통과하는 질의에 대해서만 접근을 허가하게 된다.

```
Algorithm (Semantic Policy Enforcement)
SPE (access query)
IN:   access query <S', O', A', C', P'>
OUT: PERMIT / DENY
{
      IF( ! Primitive access evaluation against <S, O, A> ∈ policy set ) {
                RETURN (DENY);
      }
      ELSE IF( ! SCCE evaluation against <S, O, C> ∈ policy set ) {
                RETURN (DENY);
      }
      RETURN (PERMIT);
}
```

그림 8. 의미적 접근제어 정책집행 알고리즘
Fig. 8. Algorithm for Semantic Enforcement of Access Control Policy

Ⅳ. 결론

새로운 정보기술의 발전으로 인해 정보 접근과 획득 방식이 훨씬 다양하고 용이해지고 있는 반면 다양하고 성능 좋은 도구를 이용한 불법적인 접근이 가능하도록 하는 부작용이 초래되고 있다.

본 논문에서는 이러한 위협에 대응할 수 있는 의미적 상황인지 접근제어 방식을 제안하였다. 본 방식에서는 보안정책에 의해 명시된 상황 제약조건의 구문이 접근질의 시 정책집행을 위한 메타정보로서 수반되는 상황제약조건의 구문과 일치하지 않은 경우에도 상황의 개념적 의미를 기반으로 의미적 정책집행이 가능하도록 하였다. 상황계층트리를 이용하여 상황개념 간의 함의관계를 형식화 하였으며 의미적 함축에 의해 초래될 수 있는 과도한 권한부여를 방지하기 위해 상황개념 간의 의미 차를 정량적으로 측정할 수 있는 SG 인수를 정의하고 이 인수를 기반으로 시스템 정의 임계치를 설정하여 제한된 범위 내에서만 함의에 의한 권한부여가 이루어지도록 하였다. 제안 방식은 정책 구문과 질의 구문의 의미적 차이를 극복할 수 있도록 하여 보안정책과 보안시스템의 관리와 운용에 있어 편이성과 효율성이 극대화 되도록 한다.

참 고 문 헌

[1] Weiser, M., "Hot Topics: Ubiquitous Computing", IEEE Computer, 1993.

[2] Kumar, N., Chafle, G., "Context Sensitivity in Role-based Access Control", Operating Systems Review, Vol. 36, No. 3, IBM Journal, 2002

[3] Wang, X.H., Xhang, D.Q., Gu, T., and Pung, H.K., "Ontology Based Context Modeling and Reasoning using OWL", in PerCom2004 Annual Conference on Pervasive computing and Communications Workshop, 2004

[4] Rastogi et al, "Access Control over Uncertain Data", PVLDB '08, 2008.

[5] P. Balbiani, "Acces control with uncertain surveillance", International Conference on Web Intelligence, 2005.

[6] Dalvi et al, "Efficient query evaluation on probabilistic databases", VLDB J, 2007.

[7] Sandhu, R., Ferraiolo, D., and Kuhm, R., "The NIST Model for Role-Based Access Control: Towards A Unified Standard", in Proceedings of the fifth ACM workshop on Role-based access control, 2000

[8] Ranganathan, R, Campbell R.H., "An Infrastructure for context-awareness based on first-order logic", Personal and Ubiquitous Computing, Vol. 7, Issue 6, 2003

[9] Qin, L., Atluri, V., "Concept-level Access Control for the Semantic Web", in ACM Workshop on XML Security, 2003.

[10] Byun, J., Bertino, E., Li, N., "Purpose-based Access Control of Complex Data for Privacy Protection", SACMAT, pp102-110, 2005

※ 이 논문은 2009년도 그리스도대학교 학술연구비 지원에 의한 논문임.

저자 소개

강 우 준(정회원)

- 1984년 2월 : 연세대학교 공과대학 전자공학과 (공학사)
- 1984년 1월 ~ 1999년 2월 : 한국 IBM 소프트웨어 연구소
- 1992년 8월 : 연세대학교 산업대학원 전자계산학 전공 (공학석사)
- 1995년 2월 : 연세대학교 경영대학원 MIS 전공 (경영학석사)
- 1999년 3월 ~ 2001년 2월 : 안산공과대학 컴퓨터공학과 교수
- 2001년 8월 : 성균관대학교 공과대학원 전기전자 및 컴퓨터 공학 전공 (공학박사)
- 2001년 3월 ~ : 그리스도대학교 경영학부 경영정보학 전공 교수

<주관심분야 : 접근제어, DRM, 전자상거래보안, XML/Web 마이닝>