프레즌스 서비스 제공을 위한 XCAP 권한관리 기법 연구*

이태진** · 김형종***

요 약

SIP 기반 프레즌스 서비스는 향후 유비쿼터스 시대의 상황 인지형 서비스에 응용될 수 있는 핵심서비스이다. 또한 프레즌스 서비스는 개인의 프라이버시 문제를 다루기 때문에, 권한관리를 위해 사용되는 XCAP 기반 기술이 최근 표준화가 진행 중에 있다. 그러나, 현재 프레즌스 기반구조를 활용한 마땅한 응용서비스 모델이 제시되어 있지 않으며, XCAP 기반 권한관리 기술이 적용된 사례가 나와 있지 않다. 본 논문에서는 새로운 프레즌스 서비스 모델을 선정하고, 서비스 개발을 위한 프레즌스 기반구조의 적용방법, XCAP 기반 권한관리 기술의 적용방안을 설계하였다. 본 논문은 프레즌스 기반 서비스 및 XCAP 기반 권한관리 기술개발을 위한 참조모델을 제안하고 있다. XCAP 기술을 통해 프레즌스 서비스의 개인정보 기능을 보강 하였고, 프레즌스 서비스의 각 메시지 전송 단계를 구체적 예를 통해 검증하였다. 본 논문의 기여점은 프라이버시를 고려한 XCAP 기반의 프레즌스 서비스 모델의 제안 및 서비스 모델 각 메시징 스텝의 검증에 있다.

An Authorization Method for Presence Service in VoIP Service

Tai Jin Lee** · Hyung Jong Kim***

ABSTRACT

Services over SIP protocol are anticipated to be commonly used services in our usual life. Especially, presence is a new feature in SIP-based services and actually entities' presence information has close relationship with privacy of them. Also, the XCAP-based authorization is accepted as a highly probable method to protect privacy of entities in SIP-based services. However, there is no proposed presence service model except IM service and it's hard to find the reference model that shows a way how we can apply XCAP-based authorization method into presence service. In this paper, we proposed new presence service model which is applicable to the VoIP service. We suggested presence service model which is making use of XCAP-based authorization to get protection of privacy in a organized way and the suggested model's each messaging steps were reviewed using concrete examples. Contributions of this work is in the suggestion of privacy-aware presence service using XCAP-based authorization and its verification of its each messaging step.

Key words : VoIP, Presence Service, XCAP, Authorization

---

* 본 논문은 연세대학교 공학대학원 석사학위 논문 초안을 기반으로 작성되었음.

** 주저자, 한국정보보호진흥원 주임연구원

*** 교신저자, 서울여자대학교 컴퓨터학부 전임강사

1. 서 론

최근 SIP을 활용한 서비스가 IETF, 3GPP등을 통해 표준화가 활발히 진행되고 있으며, 이러한 서비스 중 대표적인 것 중의 하나가 프레즌스 서비스이다. 프레즌스 서비스는 향후 유비쿼터스시대의 상황 인지형 서비스를 비롯하여 다양한 분야에서 사용될 것으로 예상된다. IETF SIMPLE WG은 프레즌스 서비스에 대한 표준화 작업을 활발히 진행하고 있다.

프레즌스 서비스의 대표적인 예는 IM(Instant Messenger)에서 Buddy의 현재 상태정보를 확인하는 것이며, 이러한 프레즌스 정보는 위치, 시간에 대한 정보 뿐 아니라 혈압 등을 체크할 수 있는 센서정보, 임의로 설정할 수 있는 상태정보 등을 포함하여, 향후 다양한 서비스가 개발될 것으로 예상된다.

한편, 프레즌스 정보는 주로 개인의 프라이버시에 관한 정보를 다루기 때문에, 악의적인 사용자에게 노출되면 다양한 피해가 발생할 수 있다. 이에 따라 프레즌스 정보에 대한 XCAP 기반 권한관리 기술이 2007년 현재 표준화가 진행 중에 있다. XCAP 기반 권한관리 기술은 프레즌스 정보를 제공하는 Presentity가 자신의 프레즌스 정보를 누구에게, 어떤 정보를 줄 것인지를 XCAP 서버에 설정한다. 이후, 프레즌스 정보를 제공받기를 원하는 Watcher의 요청이 있을 때, 프레즌스 서버는 XCAP 서버를 통해 권한이 있는지 확인하고, 권한이 있을 경우 프레즌스 정보를 Watcher에게 제공한다.

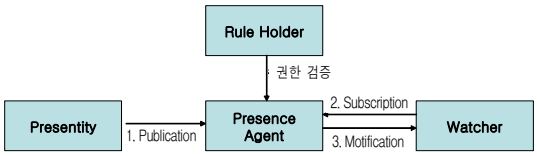
그러나, 현재 IM(Instant Messaging) 외에 마땅한 프레즌스 서비스 모델이 제시되어 있지 않고, XCAP 기반 권한 관리기술이 표준화가 추진중이나, 이에 대한 프레즌스 기반 응용서비스에의 적용 방안이 나와 있지 않다.

이러한 상황에서 본 논문은 프레즌스 서비스 모델을 활용하여 새롭게 적용할 수 있는 서비스 모델 제시한다. 또한, 제시한 프레즌스 서비스에 XCAP 기반 권한관리 기법을 적용하여 프라이버시를 보호하면서, 서비스를 제공할 수 있는 방법을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 관련 기술의 연구동향을 소개하고, 제 3장에서는 프레즌스 기반 새로운 서비스 모델을 기술하고, 제 4장에서는 서비스 개발을 위한 XCAP 권한관리기법을 설계하였다. 제 5장에서는 각 단계별 송수신 메시지를 통해 설계의 적절성을 검증하며, 제 6장에서는 결론 및 향후 연구사항을 기술한다.

2. 연구 배경

프레즌스 서비스 프레즌스 서비스 모델은 IETF SIMPLE WG에서 표준화 작업을 진행하고 있다. 프레즌스 서비스는 자신의 정보를 제공하는 Presentity와 Presentity의 정보를 얻는 Watcher, 프레즌스 정보를 관리하는 PA(Presence Agent), 권한설정을 위해 사용되는 RH(Rule Holder)로 구성된다. 프레즌스 서비스는 향후 유비쿼터스시대의 상황 인지형 서비스를 비롯한 다양한 곳에서 사용될 것으로 예상된다. (그림 1)은 IETF SIMPLE WG에서 표준화 작업을 추진하고 있는 SIP 기반 프레즌스 서비스 모델을 나타낸다.

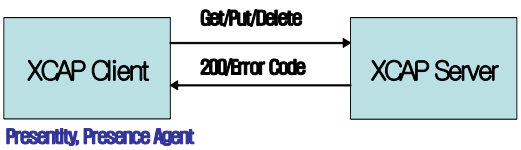


(그림 1) 프레즌스 서비스 기본 모델

Presentity는 자신의 프레즌스 정보제공을 위해 확장된 SIP PUBLISH를 사용하며, Watcher는 이 정보를 요청 및 수신하기위해 SIP SUBSCRIBE, NOTIFY 메소드를 사용한다. 한편 프레즌스 정보를 표현하기 위해 PIDF(Presence Information Data Format) 포맷을 사용하며, 다양한 프레즌스 정보를 표현하기

위해 개발된 RPID(Rich Presence Extension to the Presence Information Data Format)등을 사용할 수 있다. 또한 프레즌스 기반의 다양한 서비스를 개발하기 위해 리소스리스트를 관리하는 기법과 자신의 프레즌스 정보를 요청하는 Watcher들을 관리하는 기법 등이 표준화되어 있다[10, 11].

XCAP을 이용한 권한관리 기술 XCAP(XML Configuration Access Protocol)은 XML과 HTTP 프로토콜을 기반으로 프레즌스 정보에 대한 권한관리를 위해 사용되는 프로토콜로 IETF SIMPLE WG에서 표준화가 진행 중이다[1, 2]. XCAP Client는 자신의 프레즌스 정보를 요청하는 Watcher에 대한 권한을 생성/수정/삭제할 수 있으며, XCAP Server는 설정된 권한정보 관리 및 Watcher 요청에 응답한다. 프레즌스 기반 다양한 응용서비스는 고유한 AUID(Application Unique ID)를 통해 구분되므로, XCAP을 이용한 권한관리는 모든 응용서비스 개발에 공통적으로 활용할 수 있다[4, 5]. (그림 2)는 XCAP 권한관리 프로토콜을 나타낸다.



(그림 2) XCAP 기반 권한관리 기술

XCAP Client는 자신의 프레즌스 정보를 열람하고자 하는 Watcher들에 대한 권한설정 정보를 XCAP Server에 저장하거나 삭제, 조회를 요청하고, XCAP Server는 이에 대한 응답 메시지를 보낸다. XCAP 기반 권한관리를 위해 HTTP GET, PUT, DELETE 메소드가 사용된다.

- GET : XCAP Client 권한정보 조회
- PUT : XCAP Client 권한정보 생성 및 수정
- DELETE : XCAP Client 권한정보 삭제

〈표 1〉 XCAP 기반 권한관리 문서 포맷

```
<ruleset>
<rule id='a'>
 <conditions>
  <identity>
   <one id="sip:bob@example.com"/>
  </identity>
 </conditions>
 <actions>
  <sub-handling>Allow</sub-handling>
 </actions>
 <transformations>
   <provide-services>
 <service-uri-scheme>sip</service-uri-scheme>
<service-uri-scheme>mailto</service-uri-scheme>
   </provide-services>
   <provide-persons>
    <all-persons/>
   </provide-persons>
 </transformations>
</rule>
</ruleset>
```

XCAP을 이용한 권한관리 문서포맷 XCAP을 이용한 프레즌스 권한관리 정보는 XML로 기술되며, 기본적으로 "urn : ietf : xml : ns : pres-rules" 스키마를 따르고 있으며, "urn : ietf : params : xml : ns : com-monpolicy" 등 다른 스키마를 추가로 사용하여 세부적인 권한관리가 가능하다. XCAP을 이용한 권한관리 문서는 <표 1>과 같이 기술된다[3, 8].

<ruleset>은 권한관리에 사용되는 여러 개의 <rule>을 기술할 수 있으며, 각각의 <rule>은 <conditions>, <actions>, <transformations>로 구성되어 있다. <conditions>는 Rule이 적용되는 대상을 나타내는데, 특정대상을 지정하거나, 권한을 가질 수 있는 시간, 위치 등의 정보를 세부적으로 기술할 수 있다. <actions>는 권한을 가진 대상에 대해서 어떤 권한을 줄 것인지 기술하며, <transformations>는 해당 권한에 대한 정보 제공범위 등이 포함된다. <표 1>은 bob@example.com에게 sip, mail 등에 대한 프레즌스 정보를 제공한다는 권한이 기술되어 있다.

3. 프레즌스 서비스 모델 제안

앞서 논의한 바와 같이, 프레즌스 서비스에 대한 논의는 활발히 이루어지고 있으나, 실제 개발된 서비스 사례가 거의 없다. 여기서는 새로운 프레즌스 서비스 모델을 선정하고, 구현을 위해 필요한 XCAP 기반 권한관리기술에서의 고려사항 및 서비스 시나리오를 기술한다.

프레즌스 서비스 선정 본 논문에서 제시하는 신규 프레즌스 서비스 모델은 다음과 같다. 단말은 자신의 Buddy 목록을 볼 수 있는 화면을 가지고 있으며, 각각의 Buddy가 전화를 받을 수 있는지, 없는지에 대한 프레즌스 정보가 제공된다. 따라서 본 서비스를 통해 전화통화를 하지 않고도 상대방이 전화를 받을 수 있는지, 없는지 알 수 있는 장점이 있다. 또한, 부가기능으로 수신자가 전화를 받을 수 없는 상황일지라도, 발신자가 긴급상황일 때는 동적으로 전화를 받을 수 있는 상태로 표시되는 기능도 추가하였다.

<표 2>는 B가 A와 통화를 하지 않고도, A가 전화를 받을 수 있는지, 없는지에 대한 정보를 알 수 있는 서비스를 나타낸다. 본 서비스에서는 업무시간을 기준으로 “통화가능”, “통화불가” 정보를 설정했으나, 이는 요구하는 서비스 기능에 따라 확장하여 다양한 프레즌스 정보를 표현할 수 있다.

〈표 2〉 프레즌스 서비스 주요기능

o 대 상
A : Presentity(통화가능 여부에 대한 프레즌스 정보제공)
B : Watcher(A에게 전화를 걸기전에, A가 전화를 받을 수 있는지에 대한 프레즌스 정보 수신)

o 주요 기능
1) 업무시간에, B가 A의 프레즌스 정보를 확인할 때, “통화가능”으로 표시
2) 업무시간외에, B가 A의 프레즌스 정보를 확인할 때, “통화불가”로 표시
3) 업무시간외에, B가 긴급통화를 위해 A의 프레즌스 정보를 확인할 때, “통화가능”으로 표시

VoIP 서비스를 이용하는 사용자에게 부가서비스로 이러한 프레즌스 서비스를 제공하면, 자신의 VoIP 단말에 자주 통화하는 Buddy 목록이 있고, 각 Buddy가 현재 통화가능한지에 대한 프레즌스 정보가 제공된다. 따라서 VoIP 사용자는 자주 통화하는 Buddy 목록에 있는 사람에 대해서는 전화를 걸지 않아도 통화중인지 아닌지 알 수 있다. 물론 Buddy에 등록되어 있다고 해서 통화가용여부에 대한 정보를 모두 제공하면 프라이버시 이슈가 발생할 수 있다. 여기서는 Buddy 목록에 포함되어 있어도 XCAP을 통한 권한설정을 통해 사용자별 다른 권한을 지정할 수 있다.

프레즌스 서비스 설계를 위한 고려사항 앞서 본 논문에서 새롭게 제안한 프레즌스 서비스 모델을 선정하였다. 여기서는 실제 구현에 앞서 XCAP 기반 권한관리 기법을 적용하기 위해 다음 2가지 사항을 고려해야 한다.

(1) 프레즌스 정보 식별 : 본 서비스에서 A는 통화가능, 통화불가에 관한 상태정보, B는 긴급 상황인지 아닌지에 대한 상태정보를 제공한다. A와 B의 상태정보는 확장된 프레즌스 정보를 표현할 수 있는 RPID 스펙에 있는 <sphere> 엘리먼트를 사용한다[9, 13].

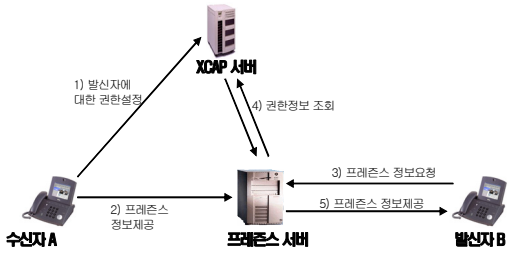
(2) 권한정보 설정 : 본 서비스에서는 업무시간과 B의 상태에 따라 A의 상태정보를 볼 수 있는 권한이 달라진다. 업무시간과 관련한 권한설정은 Common Policy 스펙에 있는 <validity> 엘리먼트를 사용하고, 프레즌스 정보 제공범위에 대해서는 Presence Auth-or-ization Rules 스펙에 있는 <provide- sphere> 엘리먼트를 사용한다. 위 시나리오에 따라 권한관리 로직을 정리하면 <표 3>과 같다[6].

〈표 3〉 VoIP 프레즌스 서비스를 위한 권한관리 로직

Watcher현재 상태 (Sphere)	정보요청 시간 (Validity)	정보 제공여부 (Actions)	제공 (Transforma tions)
normal	업무시간중	Allow	“통화가능”
normal	업무시간외	Deny	“통화불가”
emergency	any	Allow	“통화가능”

<표 3>의 권한관리 논리에 따라, Watcher의 현재상태가 emergency이면, 프레즌스 정보요청 시간과 상관없이 통화를 가능하도록 하며, normal 상태인 경우, 업무시간 여부에 따라서 프레즌스 정보를 제공할 것인지, 말 것인지를 결정한다[7].

서비스 시나리오 위 내용을 바탕으로 프레즌스 서비스 전체 시나리오를 구성하면 다음과 같다. 구성요소로는 발신자와 수신자가 있고, 프레즌스 정보를 처리하는 프레즌스 서버와 권한관리를 하는 XCAP 서버로 이루어져 있다. 본 서비스 모델은 VoIP 서비스가 제공되는 가운데, 부가서비스로 제공되는 형태를 나타내므로 VoIP 서비스망은 따로 표현하지 않고, 본 서비스 모델중심으로 표현한다.



(그림 3) VoIP 프레즌스 서비스 시나리오

(1) 수신자 A는 자신의 통화가능여부에 관한 프레즌스 정보를 볼 수 있는 대상에 대한 권한을 XCAP 서버를 통해 설정한다.

(2) A는 근무시간에는 “통화가능”, 근무시간외에는 “통화불가”의 정보를 프레즌스 서버에 보낸다.

(3) 발신자 B는 수신자 A와 통화할 수 있는지 확인하기 위해 A의 프레즌스 정보를 프레즌스 서버에게 요청한다.

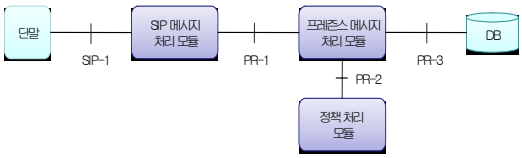
(4) 프레즌스 서버는 XCAP 서버에게 발신자 B가 권한이 있는지 확인한다.

(5) 권한이 있을 경우, 프레즌스 서버는 발신자 B에게 수신자 A의 프레즌스 정보를 제공한다.

4. XCAP 기반 권한관리 기법 설계

여기서는 앞서 선정한 서비스 모델구현을 위한 프레즌스 기반구조 및 XCAP 기반 권한관리 기법에 대한 상세설계를 기술하였다.

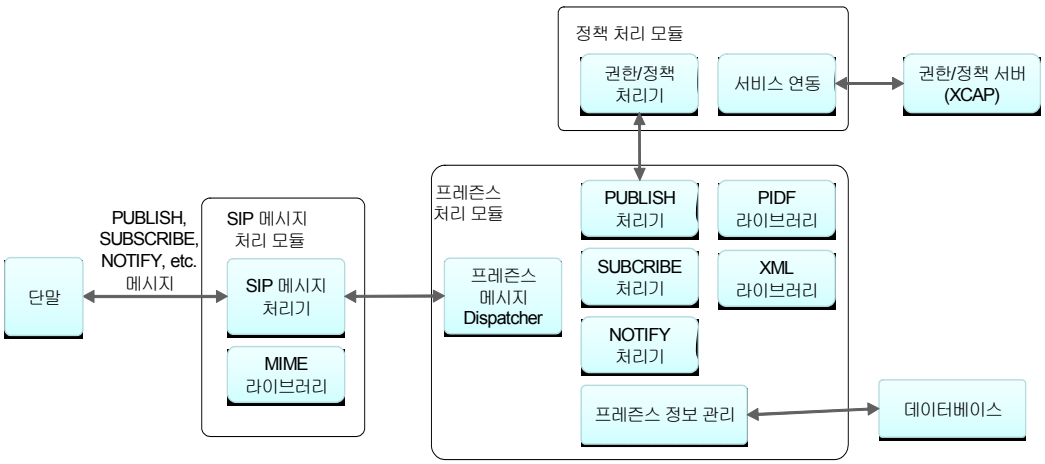
프레즌스 기반구조 설계 (그림 4)는 프레즌스 기반 안전한 서비스 제공을 위한 네트워크 구성도를 나타낸다. 클라이언트는 프레즌스 서버에 자신의 상태 정보를 전달하고 필요한 다른 단말의 상태 정보를 요청하여 전달 받을 수 있다. 이때, 사용하는 상태 정보는 클라이언트 on/off 정보, 위치 정보, id(sip url) 정보 등이 있다. 이러한 정보는 다음과 같은 시나리오로 동작한다.



(그림 4) 프레즌스 기반구조

SIP 프록시는 네트워크로부터 요청되는 SIP 메시지를 분석하여 SIP 프록시 도메인 내에 있는 장치(서버 혹은 단말) 정보를 비교하여 해당 SIP 메시지를 전달한다. 만약 도메인에 포함되어 있지 않은 SIP 메시지인 경우 해당 도메인을 검사하여 메시지를 전달(forwarding)한다. 이를 위해, SIP 프록시 도메인에 속하기 위해서는 장치는 SIP 프록시에 등록 절차를 거쳐야 하며 SIP 프록시는 등록 기능을 제공한다.

프레즌스 서버는 클라이언트(사용자)가 자신의 정보를 다른 클라이언트들에게 알리는데 이용된다. 클라이언트 정보는 프레즌스 정보(Presence Information)의 형태로 제공되고, 프레즌스 정보의 관리를 위한 구성요소들이 존재한다. 이러한 기능을 수행하기 위해서 프레즌스 서버는 Presentity와 Watcher 2개의 요소와 정보를 다루는 PUBLISH, SUBSCRIBE,



(그림 5) 프레즌스 기반구조 세부설계

NOTIFY 요소와 프레즌스 정보를 분석하고 생성하기 위한 pidf 모듈로 구성된다.

Presentity 모듈은 프레즌스 서비스의 저장 및 배포를 담당하며 Watcher 모듈은 프레즌스 정보를 수신하는 기능을 제공한다. (그림 5)는 주요 모듈이 포함된 그림이다.

SIP 메시지 처리 모듈은 SIP 메시지를 분석하거나 생성하여 SIP 프로토콜을 통해 SIP 메시지를 송/수신하는 기능을 제공한다. 프레즌스 메시지 처리 모듈은 PIDF로 표현되는 프레즌스 정보를 등록하고 정보를 요청하는 클라이언트에 따라 정보를 제공한다. 정책 처리 모듈은 클라이언트가 프레즌스 정보 열람을 요청하는 경우 어떤 권한이 있는지를 판단하고 해당 권한에 따라 관련 정보를 제공한다. 외부 서비스 서버는 정책/권한 서버를 예로 들 수 있는데, 정책 처리 모듈은 이 외부 서비스 서버에 접속하기 위한 네트워크 접속 인터페이스와 해당 정책값을 요청하고 내부 프레즌스 메시지 처리 모듈로 해당 정보를 전달한다. 각 모듈의 주요 기능은 <표 4>와 같다.

<표 5>는 주요 모듈간 참조 인터페이스를 나타낸다.

〈표 4〉 프레즌스 기반구조 주요 기능

모 듈	주요 기능
SIP 메시지 처리 모듈	· RFC 3261에서 정의하는 SIP 메시지 송/수신 · RFC 3903, RFC 3265, RFC3428 기반의 프레즌스 메시지분석 - PUBLISH, SUBSCRIBE, NOTIFY, MESSAGE, OK · 프레즌스 메시지 처리 모듈로의 메시지 처리 요청 및 응답
프레즌스메시지 처리 모듈	· 프레즌스 주요 메시지(PUBLISH, SUBSCRIBE, NOTIFY) 분석 및 생성 · 메시지별 이벤트 타입 분석 및 처리 · PIDF 형식 분석 및 생성 · 프레젠티티, 와처 관리 · 프레즌스 정보 관리
정책 처리 모듈	· 프레즌스 메시지 처리 모듈에서의 정책 요청 처리 - XCAP 관련 데이터의 경우 XCAP 서버로의 관련 정보 요청 및 수신 처리
단말	· SIP 메시지 송/수신 · SIP 메시지 분석 및 생성 · PIDF 메시지 형식 처리 · 프레즌스 정보 분석 및 생성
외부 서비스 서버	· 프레즌스 서비스를 지원하는 응용 서비스 서버(예 : XCAP)와의 연동 · 응용서비스 서버와의 연동을 위한 접속 프로토콜 지원
DB	· 프레즌스 정보 저장

〈표 5〉 프레즌스 기반구조 참조 인터페이스

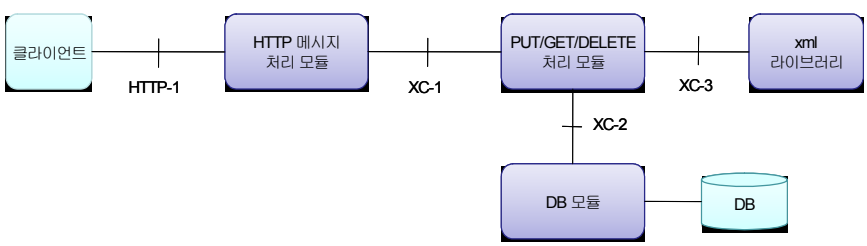
참조 인터페이스	설 명
SIP-1	· 단말-SIP 메시지 처리 모듈 간 인터페이스 · SIP 프로토콜 기반 · SIP 메시지 송/수신
PR-1	· SIP 메시지 처리 모듈-프레즌스 메시지 처리 모듈 간 인터페이스 · 함수 호출(function call) 기반 · PUBLISH, SUBSCRIBE, NOTIFY 메시지 처리 요청 및 결과 수신
PR-2	· 프레즌스 메시지 처리 모듈-정책 처리 모듈 간 인터페이스 · 함수 호출(function call) 기반 · 프레즌스 정보에 대한 권한/정책 처리 요청
PR-3	· 프레즌스 메시지 처리 모듈-데이터베이스 간 인터페이스 · 함수 호출(function call) 기반 · 프레즌스 정보 저장
EXT-1	· 외부 서비스 모듈-정책 처리 모듈 간 인터페이스 · 외부 서비스 지원 프로토콜 기반(예 : XCAP인 경우 HTTP 기반) · 외부 서비스와의 연동을 위한 프로토콜 지원 · 외부 서비스서버와의 서비스메시지 송/수신(예 : XCAP인 경우 xcap 데이터)

XCAP 기반구조 설계 프레즌스 서비스 개발을 위한 공통으로 사용되는 XCAP 기반구조는 다음과 같다. (그림 6)은 XCAP 서비스 제공을 위한 네트워크 구성도를 나타낸다. XCAP 서비스는 HTTP 기반의 프레즌스 서비스 제공 환경이다.

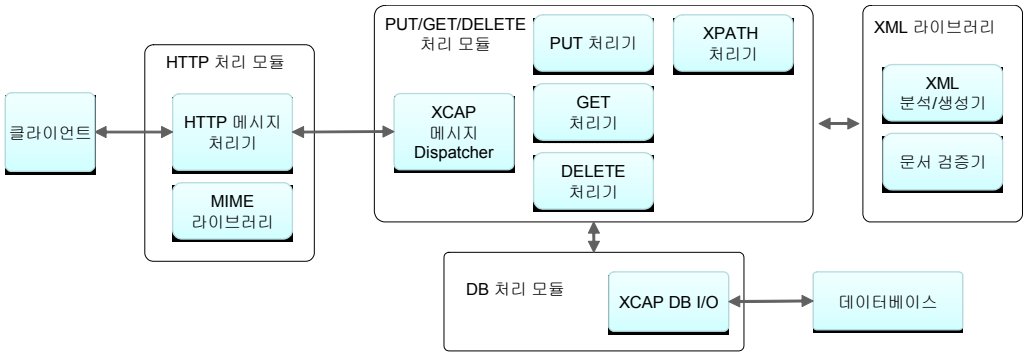
XCAP 서버는 특정 응용 서비스를 위한 클라이언트 설정 정보를 XCAP 서버에 등록해 두고, XCAP 클라이언트에서 서버에 등록된 정보를 직접 관리한다. XCAP 서버에 저장되어 있는 응용별 클라이언트 설정 정보는 XML로 기술되어 있으며, XCAP 클라이언트에서 XCAP 서버로 접근하기 위한 프로토콜로는 HTTP 1.1을 사용한다. GET, PUT, DELETE 등 3가지 HTTP 메소드를 사용하고 있으며, XML 문서 내 특정 노드를 핸들링 하기 위해 HTTP URI 매핑이 필요하다. 프레즌스 서비스를 위한 XCAP 서비스 모델은 프레즌스 정보를 해당 사용자에게 직접 통지 받는 단말간 서비스 모델과 PA 서버로부터 통지 받는 서버를 경유한 서비스 모델이 있으며, 서버를 경유한 프레즌스 서비스 모델을 기반으로 구성한다. (그림 7)은 XCAP 기반구조에 대한 세부설계를 나타낸다.

XCAP 서버가 가지는 주요 기능별 구성요소 및 인터페이스를 정의하고 있는데, XCAP 서버는 HTTP 메시지 처리 모듈, PUT/GET/DELETE 처리 모듈, XML 라이브러리, DB 모듈로 구성된다. HTTP 메시지 처리 모듈은 HTTP 1.1 기반의 메시지를 처리한다.

MIME 타입의 HTTP 메시지를 생성하거나 파싱하는 역할을 수행한다. HTTP 처리 모듈은 기본적으로 WAS(Web Server Application)에서 수행하며 본 XCAP 서버에서는 WAS에서 PUT, GET, DDELETE 메소드에 대한 요청이 들어온 경우에 해당 요청을 처리하기 위한 모듈을 호출한다.



(그림 6) XCAP 기반구조



(그림 7) XCAP 기반구조 세부설계

〈표 6〉 각 컴포넌트 주요기능

컴포넌트	주요 기능
HTTP 메시지 처리 모듈	· RFC 2616에서 정의하는 HTTP 메시지 송/수신 · PUT, GET, DELETE 메시지 분석 및 처리 모듈 호출
PUT/GET/DELETE 처리 모듈	· XPATH 처리를 위한 HTTP 메시지 헤더 분석 및 경로 계산 · XML 다큐먼트 요소 검증 및 데이터 파싱/생성 · PUT, GET, DELETE 메시지 처리
XML 라이브러리	· XML API 제공
DB	· XCAP 정보 저장

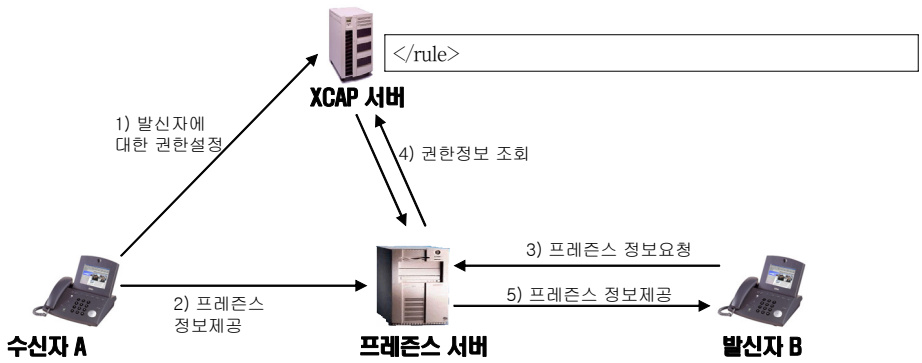
PUT/GET/DELETE 처리 모듈은 XCAP 표준에서 정의하는 PUT, GET, DELETE 메시지별 XML 문서에 대한 생성/검색/삭제 기능을 제공한다. XML 라이브러리는 XCAP 서버에서 요청받는 메시지에 대한 정보들을 처리하기 위해 필요한 경로인 XPATH 처리와 XML 다큐먼트 분석/생성에 필요한 라이브러리를 제공한다. <표 6>은 각 컴포넌트의 주요 기능을 나타낸다. <표 7>은 XCAP 모듈간 참조 인터페이스를 나타낸다.

〈표 7〉 참조 인터페이스

참조 인터페이스	설 명
HTTP-1	· 클라이언트-HTTP 메시지 처리 모듈 간 인터페이스 · HTTP 1.1 기반 메시지 송/수신
XC-1	· HTTP 메시지 처리 모듈-PUT/GET/DELETE 메시지 처리 모듈 간 인터페이스 · 함수 호출(function call) 기반 · XCAP 정의 메시지(PUT, GET, DELETE) 처리 요청 및 결과 수신
XC-2	· PUT/GET/DELETE 메시지 처리 모듈-DB 모듈 간 인터페이스 · 함수 호출(function call) 기반 · XCAP 정보 저장/획득
XC-3	· PUT/GET/DELETE 메시지 처리 모듈-XML 라이브러리 간 인터페이스 · 함수 호출(function call) 기반 · XML 분석 및 생성을 위한 API 호출

5. 제안 모델 검증

앞서, 새로운 프레즌스 서비스 모델을 선정하고, 개발을 위한 XCAP 기반구조를 설계하였다. 본 논문에서는 시나리오에 따른 주요 송수신 메시지를 대상으로 검증하고자 한다. (그림 8)은 앞서 살펴보았던 프레즌스 서비스 시나리오이다.



(그림 8) VoIP 프레즌스 서비스 시나리오

(1) 수신자 A는 자신의 통화가능여부에 관한 권한정보 설정 (XCAP을 통한 PUT 메소드 사용)

(2) 수신자 A의 프레즌스 정보 제공(프레즌스 서비스 모델을 통한 PUBLISH 메소드 사용)

(3) 발신자 B의 수신자 A에 대한 프레즌스 정보 요청(프레즌스 서비스 모델을 통한 SUBSCRIBE 메소드 사용)

(4) 프레즌스 서버는 XCAP 서버에게 권한정보 요청(XCAP을 통한 GET 메소드 사용)

(5) 프레즌스 서버는 프레즌스 정보제공(프레즌스 서비스 모델을 통한 NOTIFY 메소드 사용)

각 단계에서의 동작과정 및 송수신 메시지는 위와 같으며, 송수신 세부 메세지는 다음 단락을 통해 기술한다.

〈표 8〉 XCAP을 통한 권한설정 rule a

```
HTTP PUT
 /pres-rules/users/A/pres-rules.xml
 HTTP/1.1
 Accept : */*
 Accept-Language : ko
 If-Match : "kkll"
 Content-Type : application/pres-rules+xml
 Host : xcap.kisa.or.kr

<?xml version = "1.0" encoding = "UTF-8"?>
<ruleset xmlns = "urn : ietf : params : xml : ns : common-
                   policy"
  xmlns = "urn : ietf : params:xml:ns:pres-rules"
  xmlns = "http://www.w3.org/2001/XMLSchema-
           instance">
<rule id="a">
 <conditions>
  <sphere>emergency</sphere>
 </conditions>
 <actions>allow</actions>
 <transformations>
  <provide-sphere>TRUE</provide-sphere>
 </transformations>
```

〈표 9〉 XCAP을 통한 권한설정 rule b

```
<rule id = "b">
 <conditions>
  <validity>
    <from>2007-03-29T09 : 00 : 00 + 01 : 00</from>
    <to>2007-03-29T18 : 00 : 00 + 01 : 00</to>
  </validity>
 </conditions>
 <actions>allow</actions>
 <transformations>
   <provide-sphere>TRUE</provide-sphere>
 </transformations>
</rule>
</ruleset>
```

Watcher에 대한 XCAP 기반 권한정보 설정 우선, 프레즌스 정보를 제공하는 Presentity A는 자신에

게 전화를 걸 수 있는 B를 포함한 Watcher들에 대한 권한설정을 한다. A는 통신 프로토콜로 HTTP PUT 메소드를 사용하며, 문서포맷은 앞서 논의된 pres-rules와 common policy 스키마를 사용하여 XCAP 서버에게 전송한다. 해당화일이 없을 경우 새롭게 생성이 되며, 이미 있을 경우 기존에 설정된 권한설정 파일을 덮어쓰게 된다. 한편, 권한설정 정보를 조회할 때는 HTTP GET, 삭제할 때는 HTTP DELETE 메소드를 사용한다. 본 시나리오에 따른 XCAP 기반 권한설정 정보는 다음과 같다.

<표 8>의 첫 번째 Rule은 정보를 요청한 Watcher B의 상태정보가 emergency일 경우, Presentity A의 프레즌스 정보열람을 허용하며, A의 Sphere 정보를 제공한다는 의미이며, 표 9의 두 번째 Rule은 업무시간에 해당하는 오전 9시부터 오후 6시 사이에 A의 프레즌스 정보를 요청하는 B에게는 A의 정보열람을 허용하며, A의 Sphere 정보를 제공한다는 의미이다. 따라서 업무시간 이외에 A의 프레즌스 정보를 요청하면, 해당되는 권한이 없어 프레즌스 정보를 제공받을 수 없게 된다.

Presentity 정보를 프레즌스 서버에 제공 업무시간에 따라 Presentity A는 “통화가능”, “통화불가”에 해당하는 프레즌스 정보를 SIP PUBLISH 메소드를 통해 프레즌스 서버에게 제공한다. 프레즌스 정보는 앞서 논의된 대로, PIDF의 확장된 스펙인 RPID에서 정의하고 있는 <sphere> 엘리먼트를 사용한다.

<표 10>은 Presentity가 프레즌스 정보를 제공하는 메시지를 나타낸다. Presentity는 오전 9시부터 오후 6시까지 근무시간일 경우, PUBLISH 메소드를 사용하여, 프레즌스 정보를 프레즌스 서버에 제공한다. 이후 특정 요청자가 프레즌스 정보를 요청할 때, 적법한 요청자이면 프레즌스 서버는 프레즌스 정보를 제공하게 된다.

프레즌스 정보는 PIDF 포맷에 따라 <status> 엘리먼트 내에 상태정보를 표현할 수 있게 되어있는데, PIDF의 확장된 포맷인 RPID에 기술되어 있는 <sphere> 엘리먼트를 사용하여 프레즌스 정보를 기술하였다. 기본적인 PIDF외에 RPID가 추가 사용되었으므로, 네임스페이스에 이에 대한 정보가 추가되었다.

〈표 10〉 SIP PUBLISH를 통한 Presentity 정보 제공

```
PUBLISH sip : A@kisa.or.kr SIP/2.0
Via : SIP/2.0/TCP/ watcherhost.kisa.or.kr ; branch = xxx
From : <sip : A@kisa.or.kr> ; tag = ccc
To : <sip : A@kisa.or.kr>
Call-ID : 1111@watcherhost.kisa.or.kr
CSeq : 1 SUBSCRIBE
Max-Forwards : 70
Event : Presence
Accept : application/pidf + xml
Contact : <sip : A@kisa.or.kr>
Expires : 600
Content-Length :
<?xml version = “1.0” encoding = “UTF-8”?>
<presence xmlns = “urn : ietf : params : xml : ns : pidf”
          xmlns = “urn : ietf : params : xml : ns : pidf : rpid”>
<tuple id = “aa”>
 <status><basic>open </basic> </status>
 <rpid : sphere>통화가능</rpid : sphere>
 <contact>im : A@kisa.or.kr</contact>
</tuple>
</presence>
```

Presentity에 정보요청 Watcher B는 Presentity A에게 전화를 걸기전에, A가 전화를 받을 수 있는지 확인하기 위해 A에 대한 프레즌스 정보를 프레즌스 서버에게 요청한다. 이를 위해 SIP SUBSCRIBE 메소드를 사용한다. <표 11>은 Watcher에 의한 Presentity의 프레즌스 정보를 요청하는 SUBSCRIBE 메소드를 나타낸다.

XCAP 서버를 통한 권한정보 조회 프레즌스 서버는 Watcher B에게 Presentity A의 프레즌스 정보 제공여부를 결정하기 위해 XCAP 서버에게 문의한다. 여기서 문의하는 방법은 표준에 명시되지 않았으나, 다음과 같은 3가지 방법을 생각해볼 수 있다.

〈표 11〉 SIP SUBSCRIBE을 통한 Presentity A 정보 요청

```
SUBSCRIBE sip : B@kisa.or.kr SIP/2.0
Via :
SIP/2.0/TCP/ watcherhost.kisa.or.kr ; branch = xxx
From : <sip : Br@kisa.or.kr> ; tag = ccc
To : <sip : B@kisa.or.kr>
Call-ID : 1111@watcherhost.kisa.or.kr
CSeq : 1 SUBSCRIBE
Max-Forwards : 70
Event : Presence
Accept : application/pidf + xml
Contact : <sip : B@kisa.or.kr>
Expires : 600
Content-Length : 0
```

(1) 프레즌스 서버와 XCAP 서버가 권한관리 정보에 대한 데이터베이스를 공유하는 방법

(2) XCAP Client(Presentity)가 XCAP 서버에 설정된 권한조회를 위해 HTTP GET 메소드를 사용하는 것과 마찬가지로, 프레즌스 서버도 HTTP GET 메소드를 사용하는 방법

(3) 권한정보가 변경될때마다 XCAP 서버는 프레즌스 서버로 알려주는 방법

여기서는 Presentity가 XCAP 서버에 설정된 권한을 조회할 때 GET 메소드를 사용하는 것과 마찬가지로, 프레즌스 서버 역시 XCAP 서버에게 같은 방식인 GET 메소드를 사용하여 조회한다.

Presentity의 프레즌스 정보 제공 프레즌스 서버는 Watcher B에게 NOTIFY 메소드를 통해 Presentity A에 대한 프레즌스 정보를 제공한다.

6. 결론 및 향후연구

본 논문에서는 새로운 프레즌스 서비스 모델을 선정하고, XCAP 기반 권한관리 기술의 적용방안을 제시하였다. 본 논문의 연구 결과는 프레즌스 기반 서비스 개발 및 XCAP 기반 권한관리 기술 개발에 대한 참조모델로 활용할 수 있다. 특히, 보안에 민감한 프라이버시 정보를 다루는 프레즌스 서비스 보호를 위해 제안된 XCAP 기반기술은 유용하게 활용 될 수 있다.

향후, 성공적인 프레즌스 서비스 상용화를 위해서는 복잡한 프레즌스 정보에 대한 권한관리, 리소스리스트에 대한 권한관리, Watcher에 대해 중복된 Rule이 적용될 때의 Rule Combine 알고리즘 등에 대한 참조모델이 개발되어야 한다. 또한, 프레즌스 서비스의 개인정보보호를 위해 프레즌스 서비스에 대한 정보보호 요구사항 도출 및 보안 프레임워크 개발이 필요하다.

참 고 문 헌

[1] Wook Hyun, Sunok Park, IlJin Lee, Shingak Kang ETRI, “A Study on design and implement of XCAP Server”, Advanced Communication Technology, 2006, (ICACT 2006), Feb. 2006.

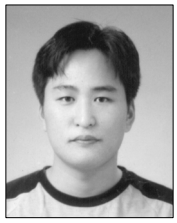
[2] J. Rosenberg, et al., “The Extensible Language(XML) Configuration Access Protocol(XCAP)”, draft-ietf-simple-xcap-05, 2004.

[3] J. Rosenberg, “Presence Authorization Rules”, draft-ietf-simple-presence-rules-01, 2004.

[4] Anand Dersingh, Ramiro Liscano, Allan Jost, “Managing Access Control for Presence-based Services”, Communication Networks and Services Research Conference, 2005 (CNSR2005), May 2005, pp. 105-111.

[5] Alam, Muhammad T. and Wu, Zheng da Bond University, “Admission control approaches in the IMS presence service”, International Journal of Computer Science,

Jan. 2006, pp. 299-314.
[6] IlJin Lee, SunOk Park, Wook Hyun, ShinGak Kang, "A Study on buddy list control in XCAP server system for SIP-based IMPP services", Advanced Communication Technology, 2006, (ICACT 2006), Feb. 2006.
[7] J. Rosenberg, "A presence Event Package for Session Initiation Protocol(SIP)", IETF RFC 3856, 2004.
[8] IETF, "Common Policy : A Document Format for Expressing Privacy Preferences", RFC 4745, 2007.
[9] IETF, "Presence Information Data Format (PIDF)", RFC 3863, 2004.
[10] 조현규, 이기수, 장춘서, "부분 Publication 및 확장 호처리언어를 사용한 새로운 SIP 프레즌스 서비스에 관한 연구", 한국콘텐츠학회 논문지 Vol.7, No. 3, pp. 34-41, 2007.
[11] "SIP 기반 프레즌스 서비스 프로파일 : 리소스 리스트 가입 및 통지", TTAS_KO-01_0076, TTA 단체표준, 2005.
[12] Matti Rantanen, "A Presence Service for Ubiquitous Computing", Mater Degree Thesis, Helsinki University of Technology, 2002.
[13] IETF, "RPID: Rich Presence Extensions to the Presence Information Data Format (PIDF)", RFC 4480, 2006.
[14] IETF, "A Session Initiation Protocol (SIP) Event Notification Extension for Resource Lists", RFC 4662, 2006.



이 태 진

2003년 포항공과대학교 컴퓨터공학과 (공학사)
2008년 연세대학교 대학원 컴퓨터공학과 석사과정
2003년~현재 한국정보보호진흥원 연구원

김 형 종

1996년 성균관대학교 정보공학과 (공학사)
1998년 성균관대학교 대학원 정보공학과 (공학석사)
2001년 성균관대학교 대학원 전기전자및컴퓨터학과 (공학박사)
2001년~2007년 한국정보보호진흥원 수석연구원
2004년~2006년 미국 카네기멜론대학 Visiting Researcher
2007년~현재 서울여자대학교 컴퓨터학부 전임강사