https://doi.org/10.7840/kics.2024.49.11.1631

데이터 공유를 위한 그룹 속성의 관리 방안 연구

조 진 용◆, 장 민 석*, 김 승 해*, 조 부 승*

A Study on the Management of Group Attributes for Data Sharing

Jinyong Jo◆, Min-seok Jang*, Seung-Hae Kim*, Buseung Cho*

요 약

e-Science는 컴퓨팅 리소스, 과학 도구 및 네트워크 기술을 활용하여 데이터의 공유와 분석을 가능하게 하는 과학적 활동으로써, 해외에서는 가상 조직을 관리할 수 있는 시스템을 개발하여 신원연합에 적용함으로써 데이터에 대한 접근성과 개방성을 높여왔다. 많은 과학기술 소프트웨어가 사용자 그룹 정보를 활용하여 데이터를 공유할 수 있는 기능을 갖추고 있지만, 국내에서는 가상 조직을 관리할 수 있는 시스템이 부족하고 그룹 정보의 이용에 대한 사용자의 경험이 부족하여 데이터의 공유 환경 조성에 어려움이 있다. 본 연구에서는 신원연합 환경에서 가상 조직에 속한 사용자들의 그룹 정보를 관리하고 그룹 정보가 포함된 인증 메시지를 중계하기 위해 개발한 속성권한 관리시스템과 인증 프록시 시스템을 제안한다. 개발한 시스템은 권한 정보의 관리를 서비스제공자에게 위임하고, 가상 조직과 서비스의 개념을 서비스 중심의 가상조직으로 통합하여 관리함으로써 시스템을 경량화하고 사용자 경험을 개선했다. 본 논문은 개발한 시스템을 운영 중인 스토리지 서비스에 적용하여 그룹 정보의 설정 방법과 전달 절차를 제시함으로써 기술 활용의 모범 사례를 제시하고 활용 가능성을 입증하였다.

키워드 : 그룹관리, 사용자 속성, 권한 관리, 인증 프록시, 신원연합

Key Words : Group management, user attribute, entitlement management, authentication proxy, identity federation

ABSTRACT

e-Science is a scientific activity that enables data sharing and analysis by utilizing computing resources, scientific tools, and network technologies. Overseas, VO (Virtual Organization) management systems have been developed and applied to identity federations, realizing data sharing and increasing data accessibility and openness. Although software already have features allowing group-based user access, in Korea, there is few systems for managing VOs and a lack of user experience in utilizing group information, making it difficult to share data. This study proposes two systems that are suitable for federation operators: an attribute authority (AA) developed to manage group information of users and an authentication proxy system to relay authentication messages containing group information. The developed AA delegates the management of permission information to service providers. We integrated VOs and Services into service-driven VOs, thereby simplifying the system architecture and improving user experience. Finally, we connected the systems to a production storage service, introducing processing flows of group information, presenting best practices for utilization, and demonstrating their feasibility.

※ 본 연구는 한국과학기술정보연구원(K24L4M1C1)의 지원으로 수행되었습니다.

◆ First Author : Korea Institute of Science and Technology Information, jiny92@kisti.re.kr, 정회원

* Korea Institute of Science and Technology Information, msjang@kisti.re.kr; shkim@kisti.re.kr; bscho@kisti.re.kr, 정회원

논문번호 : 202406-115-D-RU, Received June 10, 2024; Revised July 5, 2024; Accepted July 31, 2024

Ⅰ. 서 론

e-Science는 컴퓨팅 리소스와 과학 도구 및 네트워크 기술을 활용하여 협업, 데이터 공유, 대규모 시뮬레이션과 분석 등을 가능하게 하는 과학적 활동을 의미한다[1,2]. 기반 구축과 활용의 관점에서 볼 때, e-Science는 2000년대 그리드 컴퓨팅[3] 시대를 지나 2010년대에는 사이버인프라[4]로 진화했으며, 현재의 빅데이터 및 클라우드 컴퓨팅 시대의 기반이 되었다.

연구 조직의 연합, 과학 도구의 재사용, 연구 리소스와 데이터의 공유는 e-Science의 주요 특징으로 볼 수 있다. 예를 들어, 분산 컴퓨팅 인프라인 그리드 컴퓨팅에서 기관 간 리소스 공유를 위해 개발된 글로버스 툴킷과 가상 조직(VO, Virtual Organization)의 개념 및 기관 연합형 인증체계(Identity federation, 이하, 신원연합)는 현재까지도 사이버인프라의 주요 기반으로 작동하고 있다[5-9]. 특히, 이러한 사이버인프라 기술들은 데이터의 접근성과 개방성 및 공유 가능성을 크게 향상시키므로, 국외에서는 컴퓨팅이나 데이터 플랫폼의 구축에 광범위하게 활용되고 있다[10-12].

우리나라도 2010년 중반 국내에 구축된 신원연합[13]의 운영으로, 인증인가 기반이 갖추어지고 리소스에 대한 접근성과 개방성이 제고되고 있다. 하지만, 그리드 컴퓨팅 기반[14]이 사이버인프라로 발전하지 못하면서 VO 관리시스템 등 리소스 공유를 위한 미들웨어 기술이 축적되지 않았고 활용 시나리오의 모범 사례(Best practice)도 도출되지 못했다. 해외에서는 Openstack, nextCloud, Confluence 등과 같이 그룹 권한 관리가 가능한 다수의 소프트웨어가 VO 관리시스템과 연동되어 데이터 공유 환경을 구축하는 데 활용되고 있다.

국외 연구교육망은 그룹 정보의 관리와 활용을 위해 Grouper[15], HEXAA[9], COmanage[7]와 같은 VO 관리 시스템을 개발하여 학·연 분야 서비스 플랫폼에 적용하고 있다. 또한 CILogon[6]과 같은 인증 프록시 시스템[1] (APS, Authentication Proxy System)과 VO 관리시스템을 연동하여 그룹 정보를 제공함으로써 그룹 정보의 질의 등 서비스제공자에게 요구되는 기능을 APS에 위임하고 있다.

기존 VO 관리시스템은 종단 리소스에 대한 권한(Permission)을 직접 설정하는 방식을 취하고 있다. 사용자의 신원정보 관리에 초점을 맞춘 신원연합의 운영 환경에 적용할 경우, 이러한 접근 방식은 관리 비용의 증가를 초래한다. 즉, 중앙화된 권한 설정은 관리 작업의 복잡성을 가중시키고, 시스템 관리자의 업무량을 증대시키는 요인으로 작용할 수 있다. 또한, 해당 시스템들은 가상 조직과 서비스의 개념을 구분하여 시스템 활용의 유연성을 높였지만, 가상 조직과 서비스의 개념에 익숙하지 않은 국내 여건에서는 사용자 경험이 악화되는 문제가 있다.

본 논문은 신원연합 환경에서 VO에 속한 사용자들의 그룹 정보 관리를 위해 개발한 경량 속성권한 관리시스템(AA, Attribute Authority)의 구성요소를 살펴보고, 그룹 정보가 생성되고 획득되는 과정을 상세하게 설명한다. 또한 사용자 속성의 제어와 인증 메시지의 중계를 위해 개발한 인증 프록시 시스템(APS, Authentication Proxy System)에 관해서도 소개하며, 그룹 기반의 접근제어를 위해서 AA와 APS가 스토리지 서비스에게 그룹 정보를 중계하는 과정을 자세히 소개한다. 개발한 VO 관리시스템은 권한 정보의 관리를 서비스제공자에게 위임하고, 가상 조직과 서비스의 개념을 서비스 중심의 가상조직(Service-driven VO)으로 통합하여 시스템을 경량화하고 사용자 경험을 높였다.

본 연구의 기여점과 의의는 다음과 같다. 첫째, VO 관리 기능을 경량화하여 신원연합 환경에서도 운영할 수 있는 속성권한 관리시스템을 국내 최초로 개발했다는 점에서 의의가 있다. 둘째, 개발한 AA와 APS를 운영 중인 스토리지 서비스에 적용하여 그룹 정보를 전달하는 방법과 절차를 제시함으로써, 기술 활용의 모범 사례를 제시하였고 기술의 활용 가능성을 입증했다. 연구 결과는 향후 국내 신원연합 환경에서 효과적인 VO 관리를 위한 기반 기술로 활용될 수 있다.

본 논문은 다음과 같이 구성된다. 제2장에서 본 연구의 배경 기술과 동기를 소개하고 제3장에서 관련 연구를 살펴본다. 개발한 시스템의 세부 설계 내용과 검증 결과는 각각 제4장과 제5장에서 다룬다. 마지막으로 제제6장에서 결론을 맺는다.

Ⅱ. 배 경

본 장에서는 신원연합 환경에서 사용자 속성이 전달되는 과정과 위임형 접근제어 모델을 살펴보고 연구 동기를 소개한다.

2.1 속성의 전달 및 권한 부여

SAML(Security Assertion Markup Language) 및 OIDC(OpenID Connect)/OAuth2 표준은 인증(Authentication)과 인가(Authorization)를 물리적으로 분리하여 인증 시스템의 유연성을 높였다. 그림 1은 표

1) APS는 사용자 속성을 제어하고 인증 메시지를 중계한다.

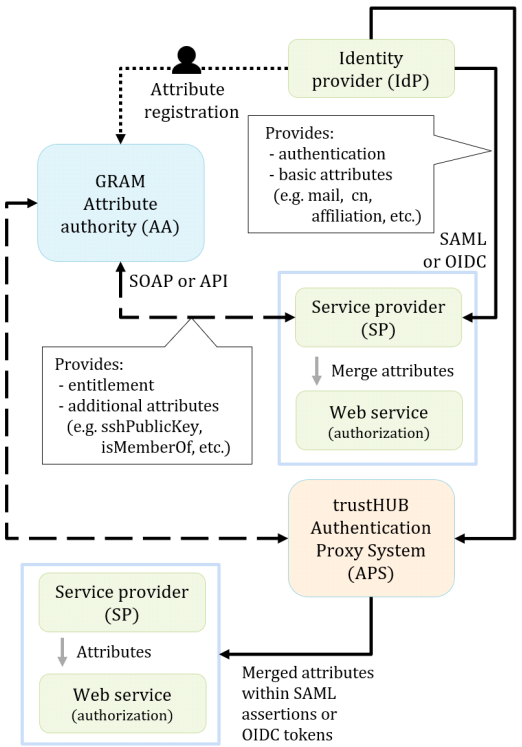

그림 1. 속성 획득 및 권한 부여 예시.
Fig. 1. Examples of acquiring attributes and granting permissions.

준 인증인가 환경에서 서비스제공자가 사용자 속성을 획득하는 두 가지 방법을 예시한다. 표준 인증인가 시스템은 사용자의 속성정보를 제공하는 아이디제공자(Identity provider)와 아이디제공자가 전달한 속성정보를 취합해 응용서비스(그림 1의 Web service)에게 전달하는 서비스제공자(Service provider)로 구분된다. 신원연합은 동일한 정책 체계를 공유하는 아이디제공자와 서비스제공자의 집합이다.

신원연합 환경에서 아이디제공자는 일반적으로 고유식별자나 mail과 같은 소속기관 구성원의 사용자 프로필 정보만 제공한다. 속성은 (a_i, V_i) 같이 특성 i를 나타내는 속성명 a_i와 속성값 v_i 또는 속성값의 집합 V_i로 표기한다. 사용자 프로필 이외의 정보 예를 들어, 여러 기관의 구성원들이 공유하는 컴퓨팅 리소스에 대한 접근 자격과 같은 부가 속성은 AA에서 제공한다. 서비스제공자는 아이디제공자에게 전달받은 사용자의 고유식별자 (a_{uid}, v_{uid})를 키값으로 해당 사용자의 자격 속성을 AA에게 요청해야 한다. 고유식별자를 일치시키기 위해 AA에 대한 사용자 인증은 서비스제공자가 인증한 아이디제공자를 통해 이루어진다.

사용자는 sshPublicKey(SSH 공개키)와 같은 부가 속성을 AA에 등록할 수 있으며, 서비스제공자는 AA로부터 부가 속성을 획득하여 자격정보(Entitlement) 등을 확인하고 리소스에 대한 접근 권한을 부여한다. 자격정보는 일반적으로 eduPersonEntitlement 속성을 이용하고 서비스제공자가 정의한 문자열을 값으로 갖는다. 예를 들어, /data 스토리지 공간에 읽기 권한만 갖는 사용자의 자격정보는 속성 $(a_{epe} = eduPersonEntitlement, v_{epe} = read:/data)$로 표시될 수 있다. 서비스제공자는 SOAP(Simple Object Access Protocol) 메시지를 활용하거나 API(Application Programming Interface)를 이용하여 AA로부터 자격정보를 획득한다.

서비스제공자는 아이디제공자와 AA에서 얻은 속성정보를 취합하여 응용서비스에 제공한다. 응용서비스는 전달받은 속성정보를 바탕으로 리소스에 대한 접근을 인가한다. 서비스제공자가 AA로부터 부가 속성을 얻으려면, AA 질의와 메시지 검증 등을 위한 추가적인 기능이 서비스제공자에 구현되어야 한다. 하지만 인증프록시 시스템(APS, Authentication Proxy System)을 사용하면 부가 속성을 얻기 위한 과정을 APS에 위임할 수 있다.

APS는 속성정보를 제어(예, 속성의 추가, 삭제, 변형 등)하고 인증 메시지를 중계하는 역할을 한다. 이를 통해 사용자들의 응용서비스 접근을 제어하고 메시지 규약이 서로 다른 인증규약 간에도 호환성을 제공한다. 또한 아이디제공자나 서비스제공자가 필요로 하는 인증인가 기능들을 탑재할 수 있다. 예를 들어, 서비스제공자나 아이디제공자는 다요소 인증(MFA, Multi-factor authentication)과 같은 보안 기능을 APS에 위임하여 공유함으로써 구현 및 운영비용을 줄일 수 있다.

2.2 위임형 접근제어 모델

신원연합에서는 응용서비스에 대한 접근관리를 AA에게 위임할 수 있다. 사용자와 리소스에 대한 접근 권한을 단일 지점에서 관리하면 운영자의 업무 부담이 줄어든다.

표 1은 신원연합에서 사용하는 위임형 접근제어 모델의 유형을 예시한다. 먼저 정책그룹 위임형(Policy group delegation) 모델은 AA에 administrator와 같은 역할 그룹(R_p, Role group)을 정의한다. 응용서비스는 정의된 역할 그룹을 표준 규약(예, SAML, OIDC/OAuth2, SOAP, LDAP, API 등)을 통해 획득하고 역할에 따른 권한(Permission, P)을 부여한다. 응용서비스가 권한을 사용자별로 구분((U, P))하면 동일한

표 1. 위임형 접근제어 모델
Table 1. Delegated access control model

Delegation model	Description
Policy group	$R_p \rightarrow P$
	$R_p \rightarrow (U,P)$
Reference attribute	$A_r \rightarrow P$
External permission	$(R_p, U, P) \rightarrow P$

역할을 갖는 사용자들에게 서로 다른 권한을 부여할 수 있다.

기준속성(Reference attribute) 위임형 모델은 사용자의 기준속성(A_r)을 지정해 권한을 부여하는 방식이다. 개별 사용자에게 특정 속성값(예, student@ex.com)을 할당하고 해당 속성을 갖는 사용자들에게 동일한 권한을 부여하는 방식이다. 일반적으로 아이디제공자 또는 AA가 기준속성을 제공하지만, 사용할 수 있는 기준속성의 수와 값이 제한되어 있어 정교한 권한 부여에 한계가 있다.

마지막으로 권한 위임형(External permission) 모델은 (U, R_p, P)를 AA에서 설정하고 응용서비스에게 P를 프로비저닝(Provisioning)하는 방식이다. 즉, 응용서비스가 역할 기반 접근제어(RBAC, Role-Based Access Control)를 수행하는 데 필요한 역할 정보를 AA로부터 받아서 활용한다. 응용서비스의 권한 관리와 AA의 권한 설정이 동기화되어야 한다.

본 연구에서 개발한 AA는 정책그룹 및 기준속성 위임형 모델을 참조하였지만, 권한 위임형 모델은 배제하고 구현되었다. 첫째, 다수의 국내 연구개발 인프라는 응용서비스가 지역적으로 사용자 권한을 관리하기 때문에 권한 위임형 모델에 대한 요구가 크지 않다. 응용서비스에서 권한을 각자 관리하게 되면, 권한 정보가 여러 서비스에 중복되어 저장되고 관리 주체가 다른 응용서비스 간의 통합이 어려워지므로, 인프라의 확장성이 제한되는 문제가 발생한다.

둘째, 인증 메시지의 제어와 중계가 주목적인 신원연합에서 신원연합 운영자가 종단 응용서비스의 권한을 직접 관리하면 인프라의 운영비용이 크게 상승한다. 예를 들어, SSH를 통해 접근해야 하는 터미널 응용은 일반적으로 LDAP을 통해 권한 관리가 이루어진다. 중앙화된 LDAP 서버와 개별 기관에서 운영하는 터미널 응용을 연동하기 위해서는 VPN(Virtual Private Network) 등 추가적인 보안 조치가 필요하다.

2.3 연구 동기

국가과학기술연구망(KREONET)은 연구자들의 협업 연구를 돕고 연구 생산성을 높이기 위해 다기능 고성능 네트워크 서비스를 제공하고 있다. 또한 연구자들이 활용할 수 있는 신원연합, 멀티미디어, 공동 작업, 데이터 저장과 공유 등의 협업 응용서비스도 제공함으로써 연구 활동을 지원하고 있다. 개인형 저장소와 같은 스토리지 서비스는 VO 형태의 조직 구성을 갖는 예를 들어, 특정 연구를 위해 여러 연구원에 소속된 개별 연구자들이 하나의 연구팀을 이룰 때, 물리적으로 분산된 연구자들 간에 데이터를 효과적으로 공유하는 데 유용하게 활용될 수 있다.

접근 권한이 관리되지 않는다면, 스토리지 공유 기능으로 인해 데이터가 유출될 수 있으므로, 스토리지 서비스는 공유 기능을 비활성화한 상태로 운영되었다. 그러나 최근 융합 연구 사업 등 VO 형태의 조직에서 데이터 공유의 필요성이 대두되었다. 해당 연구자들의 요구를 수용하는 동시에 보안 문제도 해소하기 위해, 두 가지 방안이 제안되었다. 첫째, MFA을 적용해 침해된(Compromised) 사용자의 서비스 접근을 차단한다. 둘째, 서비스 운영자가 특정 정책그룹을 구성하고 해당 그룹에 속한 사용자들끼리만 데이터 공유가 가능하도록 제한한다.

다단계 인증을 활성화하기 위해 MFA 표준 프로파일[16]을 지원하도록 OTP(One-Time Password) 소프트웨어[17]를 개발하고 운영 중인 APS에 탑재[8]했다. 본 연구를 통해 개발한 AA는 OTP 계정을 등록할 때 사용자 신원을 확인하는 수단으로 활용될 수 있지만, 본 논문에서는 해당 내용을 다루지 않는다.

승인된 그룹 사용자들 간의 데이터 공유를 가능하게 하도록 스토리지 서비스를 분석한 결과, 정책그룹 위임형 모델($R_p \rightarrow (U,P)$)을 채택하고 시스템 설정을 통해 동일한 그룹에 속한 사용자들에게 데이터의 공유 권한을 부여할 수 있음이 확인되었다. 즉, 스토리지 서비스에 로그인한 사용자의 그룹 정보를 AA로부터 얻을 수 있다면, 동일한 그룹에 속한 사용자 사이에서 데이터 공유가 가능해진다.

앞서 기술했듯이 응용서비스가 AA로부터 그룹 정보를 얻기 위해서는 SOAP 메시지나 AA가 제공하는 API를 이용해야 한다. 이는 응용서비스를 구동하는 소프트웨어의 수정을 의미한다. 그룹 정보를 활용하는 응용서비스의 수가 증가하면, 다수의 소프트웨어를 수정해야 하는 문제가 발생한다. 응용서비스가 소프트웨어를 수정하지 않아도 그룹 정보를 얻을 수 있도록, APS를 응

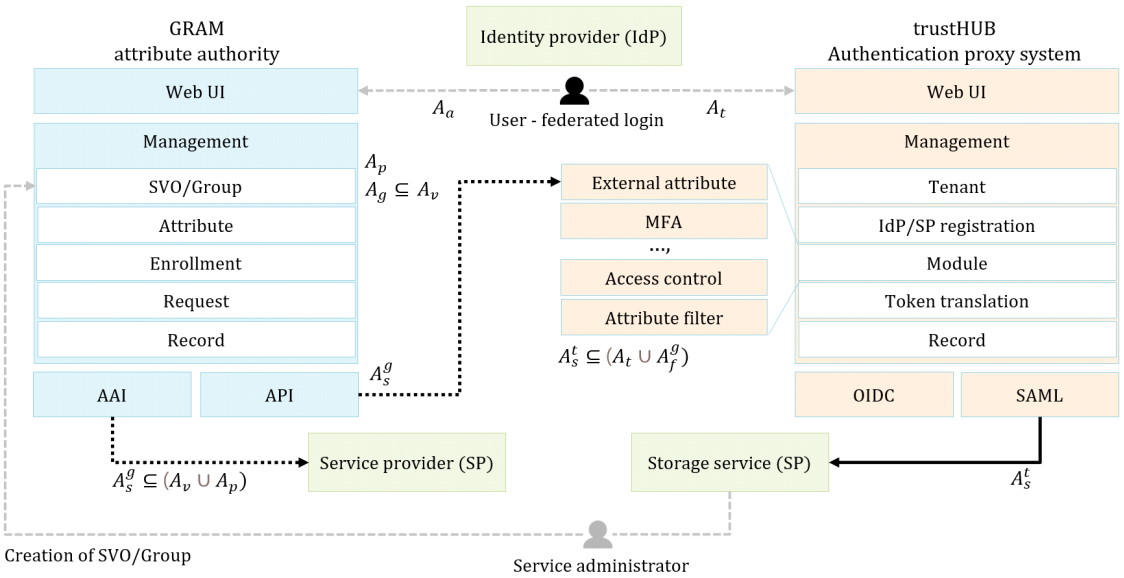

그림 2. 개발된 AA와 APS 구조의 개략도.
Fig. 2. High level overview of developed AA and APS architecture.

용서비스의 대리자로 활용하는 방안이 고안되었다.

Ⅲ. 관련 연구

본 장에서는 AA와 APS의 유사 연구를 살펴본다. 다수의 AA가 존재하지만, 신원연합에서 활용도가 높은 소프트웨어만 소개한다.

VOMS(Virtual Organization Management System[18])는 그리드 컴퓨팅 환경에서 VO에 포함된 사용자의 역할과 권한을 관리하기 위한 시스템이다. X.509 인증서로부터 신원정보를 수집하거나 역할과 권한 정보를 X.509 인증서에 기록할 수 있다. VOMS는 그리드 컴퓨팅 환경에 특화되어 있고 APS와의 연동 방식을 고려하지 않았다는 점에서 본 연구와 차이가 있다.

Internet2 Grouper[15]는 사용자의 그룹, 역할, 권한 및 접근제어를 관리하기 위한 소프트웨어이다. 신원 관리시스템(IDM, Identity Management System)과 연계되어 활용되며, 그룹과 권한 정보를 저장하기 위해 LDAP 디렉토리를 이용한다. 서비스제공자는 VOOT[19], SOAP 및 API를 이용해 Grouper로부터 그룹 정보를 얻을 수 있다. Internet2 COmanage[7]는 사용자 신원정보의 수명주기(Lifecycle)를 관리하기 위한 소프트웨어이다. 플러그인(Plugin)을 통해 LDAP과 같은 외부 소프트웨어나 서비스를 연동할 수 있어 높은 확장성을 갖는다. COmanage는 $(R_p, U, P) \rightarrow P$ 모델의 R_p와 P를 관리하기 위해 Grouper를 연동해 활용한다.

HEXAA[9]는 Grouper와 유사한 기능을 제공하지만, 새로운 속성의 등록과 관리가 유연하고 사용자 동의 기능을 제공한다는 점에서 Grouper와 차이가 있다. 본 연구는 HEXAA를 참조 모델로 하여, GRAM(Group Attribute Management)이라는 AA를 개발했다. VO 개체(E^v)와 서비스 개체(E^s)를 SVO(Service-driven VO, E_v^s)로 통합했다는 점과 AA에서 권한 P를 관리하지 않는다는 점에서 Grouper나 HEXAA와 차별화된다. 소프트웨어의 적용 범위를 제한할 수 있으나, 기능을 단순화함으로써 이용성(Usability)이 높아진다. 신원연합의 운영 측면에서 P의 관리가 비용의 증가를 의미하기 때문에, 이용성을 높이는 방향으로 개발했다.

HEXAA는 신원연합에 참여한 서비스제공자만 E^s를 등록할 수 있지만, GRAM은 시스템 관리자(S)가 승인한 모든 서비스제공자가 E_v^s를 등록할 수 있다. 또한 보안 추적성을 높이기 위해 사용자의 모든 서비스 이용 이력(예, 서비스나 속성의 생성, 삭제, 이전, 승인, 동의, 참조 등)을 관리한다는 점에서 HEXAA나 Grouper 시스템과 차이가 있다.

CILogon[6]은 APS의 일종으로 SaaS(Software as a Service) 형태의 유료 서비스이다. 본 연구에서 활용한 trustHUB APS와 유사한 기능과 구조를 갖는다. 이종 인증 규약 간에 토큰 변환(Token translation) 서비스를

제공하고 Grouper를 연계한 COmanage를 연동함으로써 $(R_p, U, P) \rightarrow P$의 권한 관리가 가능하다. CILogon의 구조와 기능을 상세히 다룬 논문을 찾기 어려워 직접적인 비교는 어렵다. 하지만 trustHUB는 X.509 인증 기능을 제외함으로써 기능을 단순화했으며, 신원연합 내에서 사용자, 서비스제공자, 아이디제공자에 대한 세밀한 접근제어가 가능하다는 점에서 본 논문은 CILogon과 차별점이 있다.

Ⅳ. 설계 및 구현

본 장에서는 GRAM과 trustHUB의 구조를 살펴보고, 각 시스템의 구성요소에서 속성이 처리되는 과정을 설명한다. 특히 속성 및 의 관리, API를 활용한 GRAM과 trustHUB의 연동 방법에 대해서 자세히 살펴본다. trustHUB에 대해서는 GRAM과의 연동을 목적으로 새롭게 개발된 부분을 위주로 설명한다. 표 2는 본 장에서 사용하는 기호를 설명한다.

표 2. 기호 및 설명
Table 2. Notation and description

Notation	Description
V	Set of service-driven VOs
G	Set of groups
R	Set of roles
P	Set of permissions
P_s	Permissions for s (a service provider)
E^v	entity of VO
E^s	entity of Service
V_s	SVO for s
G_s	Set of groups included in V_s
S	Administrator of AA
S_v	Administrator of V_s
S_g	Administrator of a group $g \in G_s$
S_s	Administrator of s
A_t	Attributes delivered to APS
A_a	Attributes delivered to AA
A_g	Attributes defined for a group $g \in G_s$
A_p	Attributes defined for user profile
A_v	Attributes defined for V_s
A_s^g	Attributes from AA for s
A_s^t	Attributes from APS for s
T_s	Tenant relevant to s

4.1 시스템 구조 및 그룹 정보의 전달

같은 그룹에 속한 사용자들에게 데이터의 공유 권한을 부여하기 위해서 그림 2와 같이 GRAM과 trustHUB를 연동했다. 앞서 기술했듯이 본 연구에서는 스토리지 서비스에 대한 접근제어를 강화하는 방안으로 MFA를 적용하고, 정책그룹 위임 방식의 접근제어 모델을 채택했다. MFA가 필요하지 않거나 응용서비스를 수정할 수 있는 경우, 그림 2의 좌측 하단에 위치한 서비스제공자와 같이 trustHUB를 거치지 않고 GRAM과 직접 연계하여 그룹 정보를 얻어올 수 있다. 아이디제공자가 MFA를 제공해야 하지만, 국내 신원연합에서는 MFA를 지원하지 않는 아이디제공자가 다수 존재하므로 MFA의 실행을 위임하기 위한 목적으로 trustHUB를 사용했다.

trustHUB는 토큰 변환(그림 2의 Token translation)을 통해 이종 규약을 사용하는 아이디제공자나 서비스제공자를 수용할 수 있다. 개별 서비스제공자의 관리자(이하 S_s)가 그림 2의 모듈(Module)에서 제공하는 기능을 직접 설정하여 이용할 수 있도록, 테넌트(Tenant) 관리기능을 제공한다. 테넌트 관리를 위해, 관리 대상이 되는 서비스제공자를 등록(그림 2의 IdP/SP registration)할 수 있다. 또한 OIDC 기반의 소셜 아이디제공자도 연동할 수 있도록 설계되었다. 소프트웨어를 모듈화함으로써, 시스템의 확장 가능성을 높였다.

GRAM과 같은 외부 AA로부터 속성정보를 얻어(External attribute)오거나, MFA를 실행하고, 사용자의 접근을 제어하며, 서비스제공자에게 전달되는 속성을 선별적으로 필터링(그림 2의 Attribute filter) 하는 등의 기능을 수행하는데 개발된 모듈들이 활용된다. 보안 사고 추적과 개인정보보호를 위해 사용자 동의, 테넌트 생성 및 삭제 등 trustHUB에서 발생하는 모든 이벤트는 기록(그림 2의 Record)된다.

GRAM은 그룹 정보와 같이 아이디제공자가 직접 관리하지 않는 사용자 속성을 서비스제공자에게 제공한다. 서비스제공자 s에서 사용할 속성을 V_s에서 관리해야 하므로, 일반적으로 S_s와 V_s의 관리자는 동일하다 ($S_s = S_v$). S_v는 V_s내에 그룹 G_s를 생성하고 개별 그룹($g \in G_s$)의 관리자(S_g)를 설정할 수 있다. 또한 S_v는 서비스제공자에게 전달할 속성을 선택한다. 속성 활용의 유연성을 높이기 위해, GRAM의 시스템 관리자(S)가 속성 관리기능(그림 2의 Attribute)을 이용해 새로운 속성을 정의하거나 속성값의 할당 방식을 수정할 수 있도

록 설계했다. S가 정의한 속성은 GRAM의 사용자 프로파일과 V 및 G에서 사용된다.

S_v 또는 S_g가 사용자를 V_s 또는 G_s에 등록하기 위해 다음 3가지 사용자 등록(Enrollment) 방식을 이용할 수 있도록 개발하여, 관리 및 이용 편의성을 높였다: 1) S_v 또는 S_g에 의한 할당, 2) S_v 또는 S_g에 의한 사용자 초대, 3) 사용자 요청에 따른 S_v 또는 S_g의 승인. 하나의 $g \in G_s$에 속한 사용자들은 동일한 그룹 식별자($i_g \in A_g$)를 부여받는다. 사용자 요청과 승인 기능(그림 2의 Request)은 관리자의 승인 권한을 차등화하고 권한을 갖는 관리자에게 사용자 요청을 라우팅하는 역할을 한다. 요청과 승인 결과는 GRAM 내부의 알림 기능과 전자우편을 통해 전달된다. 마지막으로 trustHUB와 마찬가지로 GRAM 내부의 모든 이벤트는 기록(Record)된다.

서비스제공자 s가 필요로 하는 속성 A_s^t 또는 A_s^g를 제공하기 위해서 GRAM과 trustHUB에 다음과 같은 환경이 구성되어야 한다. S_s는 trustHUB에 관리 테넌트 T_s를 생성하여 서비스제공자가 필요로 하는 속성 A_s^t를 등록한다. 또한 GRAM에 V_s와 G_s를 생성하여 사용자를 V_s와 G_s에 할당하고 trustHUB에 전달할 속성 A_s^g를 설정한다. GRAM과 trustHUB는 개인정보의 사용을 최소화하기 위해 $A_s^g \subseteq (A_g \cup A_s)$ 또는 $A_s^t \subseteq (A_t \cup A_s^g)$인 속성만 서비스제공자에게 제공한다.

사용자는 GRAM에 로그인한 후, 자신에게 설정 권한이 있는 속성값(예: 전화번호 등)을 직접 등록할 수 있다. 마지막으로 T_s는 GRAM이 제공하는 API나 AAI(Attribute Authority Interface)를 이용해 A_s^g를 얻을 수 있도록 V_s를 연계해야 한다. T_s는 V_s의 고유 식별자를 이용해 V_s를 연계하고, V_s가 제공하는 토큰을 이용해 API에 대한 접근 권한을 획득한다.

필요한 환경 설정을 마친 후, 사용자가 응용서비스에 로그인할 때 그룹 정보가 서비스제공자에게 전달된다. 그림 2의 스토리지 서비스에서 사용자가 로그인을 요청하면, 인증요청 메시지는 trustHUB를 경유하여 아이디 제공자에게 전달된다. 사용자가 아이디제공자에서 로그인에 성공하면, 해당 사용자의 속성 A_t가 인증응답 메시지에 포함되어 trustHUB로 반환된다. trustHUB에 등록된 T_s는 V_s의 연계 여부를 확인하고, API를 통해 GRAM에 해당 사용자의 그룹 정보를 요청한다. 사용자가 V_s에 참여 중이면 GRAM은 A_s^g를 반환하고, T_s는 반환된 A_s^g와 A_t를 취합한다. T_s는 취합된 속성 중 $A_s^t \subseteq (A_t \cup A_s^g)$만 스토리지 서비스에 전달한다. 마지막으로 스토리지 서비스는 A_s^t에 포함된 속성값을 이용해 리소스에 대한 접근 권한을 부여한다.

4.2 가상 조직 및 사용자 속성의 관리

본 절은 GRAM에서 관리하는 SVO와 속성의 계층적 구조에 관하여 설명한다. 앞서 기술했듯이 본 연구는 E^v와 E^s를 E_v^s로 통합했으며 P에 관여하지 않도록 AA를 설계했다. 일반적으로 E^v와 E^s는 각각 그룹 G(또는 역할 R)와 권한 P를 관리한다. E^s는 E^v가 관리하는 G를 가져와 P에 할당함으로써, 사용자 그룹에 특정 권한을 부여할 수 있다. 예를 들어, P와 G가 각각 두 개의 권한과 그룹을 갖는다면($P = \{p_1, p_2\}$, $G = \{g_1, g_2\}$), E^s는 그룹 g_1과 g_2에 속한 사용자에게 특정 권한 p_1을 부여할 수 있다($p_1 \leftarrow \{g_1, g_2\}$). 본 연구의 초기 프로토타입은 E^v와 E^s를 구분하고 P를 관리할 수 있도록 개발되었다. 하지만 G, P의 개념과 연계 방법이 국내 연구자들에게 생소해 이용성이 떨어지고, 적용할 수 있는 용례(Use case)를 찾기 어려워 V로 통합하여 기능을 간소화했다.

그림 3의 사용례에서 볼 수 있듯이 하나의 V_s는 하나 이상의 서비스제공자와 연계되고, $G_{s1} = \{g_1, g_2\}$과 같이 여러 개의 그룹을 가질 수 있다. 서비스제공자는 V_s가 부여한 URI(Uniform Resource Identifier) 형태의 고유식별자를 이용해 V_s를 연동한다. V_s에 속한 개별 그룹도 고유한 namespace를 갖는다. namespace는 역할 그룹 R_P(예, admin)를 정의하는 데 사용되도록 설계했으므로, namespace의 값은 그룹 정보를 나타내는 속성값으로 사용된다. 본 연구에서는 그룹 정보를 나타내기 위해서 eduPersonEntitlement와 isMemberOf 속성을 사용했다[2].

소프트웨어 구조를 단순화하기 위해 G_s는 평면 그룹(Flat group)만 갖도록 설계했다. 즉, 하나의 그룹은 하위 그룹을 포함하는 중첩 구조를 지원하지 않는다. 하지만 namespace를 활용하면 논리적 중첩 그룹(Logical subgroup)을 구성할 수 있다. 예를 들어, 그림 3에서 Group 1-1의 namespace를 g1/g1-1로 지정하면, Group

2) eduPersonEntitlement과 isMemberOf는 각각 사용자의 권한이나 자격을 나타내거나 사용자가 속한 그룹을 나타내기 위한 속성이다.

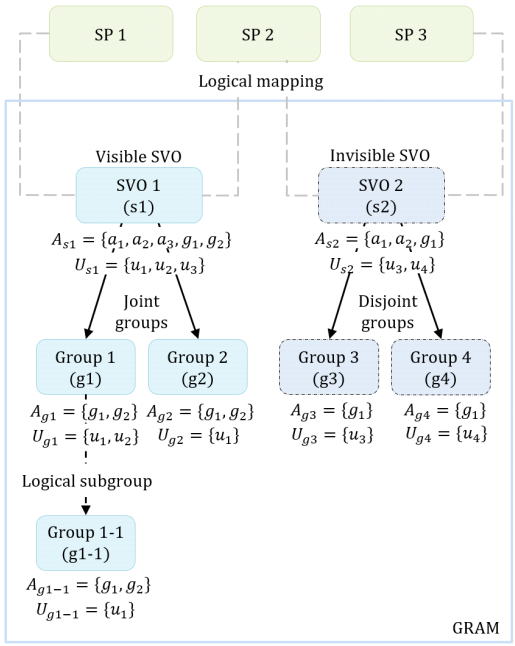

그림 3. 사용자와 그룹 및 속성의 사용례.
Fig. 3. Use cases for users, groups, and attributes.

1-1은 namespace가 g1인 Group 1의 논리적 중첩 그룹이 된다.

S_s는 G_s를 합동그룹(Joint group) 또는 배반그룹(Disjoint group)의 형태로 구성할 수 있다. 이를 통해 서비스제공자는 권한 관리 시나리오에 맞춰 합동그룹과 배반그룹을 유연하게 활용할 수 있다. 예를 들어, 한 사용자에게 '읽기 전용 권한'과 '쓰기 권한'을 동시에 부여할 수는 없으므로, 해당 시나리오는 배반그룹을 활용하는 것이 적합하다. 그림 3의 사용자 u_1과 같이 합동그룹에서는 한 명의 사용자가 여러 그룹에 참가할 수 있다. Group 3과 Group 4는 배반그룹으로써 한 사용자는 하나의 그룹($U_{g3} \cap U_{g4} = \varnothing$)에만 참여해야 한다.

V_s는 역할 그룹을 의미하지 않으므로 그룹 정보를 갖지 않는다. 즉, V_s에 속한 사용자가 그룹 정보를 갖기 위해서는 G_s에 참여해야 한다. 특정 사용자가 서비스제공자에 접근하는 것을 차단(예, 그룹 속성을 갖지 않는 사용자를 차단)하기 위한 목적으로만 V_s가 사용된다면, $|G_s| = 1$일 수 있다. V_s 보다 G_s에 먼저 참여한 사용자는 V_s에 자동으로 등록되게 개발했지만, V_s에 먼저 참여한 사용자는 직접 G_s에 참여하거나 S_s가 해당 사용자를 G_s에 포함해야 하므로 이용성과 관리 편의성이 낮아질 수 있다. S_s가 생성한 그룹을 기본 그룹(Default group)으로 지정하면, V_s에 참가하는 사용자가 자동으로 기본 그룹에 포함되도록 하여 이용성과 관리 편의성을 높였다.

V_s와 G_s의 정보가 사용자들에게 노출되는 것을 막을 필요가 있을 때, V_s의 가시성(Visibility)을 설정할 수 있도록 개발했다. G_s는 V_s의 가시성을 상속받는다. 예를 들어, 사용자 등록 방식 중 사용자 요청과 S_v 또는 S_g에 의한 승인 방식은 사용자가 직접 V_s를 선택해야 하므로 V_s 정보가 개방되어야 한다. 사용자는 공유키를 이용해 숨겨진(Invisible) V_s에 접근할 수 있다. 마지막으로 초대에 의한 사용자 등록 방식을 지원하기 위해 V와 G에 속한 모든 개체는 독립적인 초대 URL을 갖도록 설계했다.

속성은 관리 위치에 따라 전역 속성과 지역 속성으로 구분했다. 전역 속성은 V에서 공통으로 사용할 수 있는 속성(예, 전화번호 등)으로 사용자 프로필에서 관리한다. 지역 속성은 서비스제공자 s에서 필요로 하는 속성으로 V_s에서 관리한다. 지역 속성은 일반 속성(그림 2의 a_1, a_2, a_3)과 그룹 속성(그림 2의 g_1과 g_2)으로 구분되고 G_s는 V_s에 정의된 모든 그룹 속성을 갖지만, 일반 속성은 관리하지 않는다. G_s가 V_s와 동일한 일반 속성을 가지면 G_s와 V_s에서 다르게 입력한 속성값으로 인해 권한 관리에 오류가 발생할 수 있다. 마지막으로 일반 속성과 그룹 속성의 값은 각각 V_s와 G_s에서 할당하도록 설계했다.

속성값의 설정 방식은 3가지로 구분했다: 1) 사용자가 직접 입력, 2) 스크립트 코드가 자동 할당, 3) S_v가 직접 입력. 일반적으로 전역 속성은 사용자가 값을 직접 입력한다. 지역 속성 중 그룹 속성은 스크립트 코드를 통해 값을 할당하는 방식을 취했고, 일반 속성은 S_s가 값을 할당하도록 설계했다. 지역 속성은 서비스제공자가 리소스에 대한 권한을 부여하는 데 사용되는 정보이므로, 사용자가 속성값을 직접 설정하도록 허용하면, 리소스에 대한 비정상적인 접근을 차단하기 어려워진다. 스크립트 코드를 통해 속성값을 자동으로 설정할 수 있도록 GUI(Graphical User Interface) 환경이 개발되었다.

그림 4는 isMemberOf 속성의 값을 자동으로 생성하기 위해 GUI 환경에서 작성한 스크립트 코드이다. V_s에 포함된 G_s를 읽어와 사용자의 역할이 admin이거나 user이면 그룹의 namespace 값을 isMemberOf의 속성

```
def generate(wiz, service, group, config):
    svcinfo = service.info()
    if group is None:
        groups = service.group.list()
    else:
        groups = [group.info()]
    ismemberof = []
    for g in groups:
        group = service.group(g["id"])
        role = group.permission.role(strict=True)
        if role not in ["admin", "user"]: continue
        ismemberof.append(g["name"])
    return ismemberof
```

그림 4. isMemberof 값 할당을 위한 스크립트 코드.
Fig. 4. Script code for assigning isMemberOf values.

값으로 추가한다. 사용자의 역할이 null이면 즉, 그룹에 참가하고 있지 않으면 isMemberOf 값을 할당하지 않는다.

서비스제공자 s가 API를 이용해 부가 속성을 얻으려면, V_s의 고유식별자와 사용자의 고유식별자를 킷값(예, 그림 3의 $\{s1, u_1\}$)으로 사용하여 GRAM에 질의해야 한다. GRAM은 A_s^g에 정의된 속성을 사용자 프로필과 V_s 및 G_s에서 수집해 서비스제공자에게 반환한다.

4.3 속성 질의 인터페이스

속성 질의 인터페이스(Attribute Query Interface)는 그림 2의 AAI와 API를 의미한다. 서비스제공자가 사용자의 부가 속성을 질의하기 위해서는 GRAM에서 사용되는 - SAML 속성규격과 호환되는 - 사용자의 고유식별자를 알아야 한다. 비표준 인증 규약을 사용하는 응용서비스는 해당 식별자의 값을 알 수 없으므로, GRAM과 연동될 수 없다. 그림 2에서 아이디제공자가 전달한 사용자 속성에는 GRAM에 등록된 사용자의 고유식별자 $a_u \in \{A_a \cap A_t\}$가 포함되어 있으므로, trustHUB는 a_u를 킷값으로 GRAM에 질의할 수 있다.

그림 5는 SAML 서비스제공자를 대상으로 제공되는 AttributeQuery[20]의 사용례로써 통신규약은 SOAP을 사용한다. AttributeQuery는 그림 2의 AAI에 해당한다. 서비스제공자 https://sp.ex.com이 a_u가 gdhong@ex.com인 사용자의 속성 urn:oid:1.3.6.1.4.1.5923.1.1.1.7 (eduPersonEntitlement)을 요청하고 있다. AttributeQuery를 활용하려면 서비스제공자와 AA 간에 SAML 메타데이터를 교환해야 하는 등 관리가 까다

```
<?xml version="1.0"?>
<samlp:AttributeQuery
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
ID="21df07a1f85cffed5797d1fa84b10bab"
Version="2.0" IssueInstant="2024-05-27T16:14:53Z">
  <saml:Issuer
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
      https://sp.ex.com
  </saml:Issuer>
  <saml:Subject
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
 <saml:NameID>gdhong@ex.com</saml:NameID>
  </saml:Subject>
  <saml:Attribute
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname
-format:uri" Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7"/>
</samlp:AttributeQuery>
```

그림 5. AttributeQuery의 예.
Fig. 5. Example AttributeQuery.

로우며, SAML 이외의 인증 규약에서는 AttributeQuery를 사용할 수 없다는 문제가 있다. 이 문제를 해결하기 위해 create, list, join, status 등 총 12개의 API를 개발하고 토큰을 이용해 API에 접근할 수 있도록 하여 보안성을 높였다. 또한 서비스 거부 공격을 방지하기 위해, 일정 시간 동안 호출할 수 있는 API 횟수를 제한하도록 설계했다.

그림 6은 사용자의 부가 속성을 얻기 위한 속성 질의 API와 질의 결과를 예시한다. 사용자의 고유식별자인 gdhong@ex.com과 V_s의 고유식별자인 gramtest, API 토큰값, 서비스제공자가 필요로 하는 속성을 조합하여 https://<[URL Prefix>/service/user/attribute로 질의했다. 질의 결과(그림 6의 Response)는 JSON(JavaScript Object Notation) 형태이며, 질의 시 요구한 속성명과 속성값을 포함해 반환된다.

```
Attribute Query API:
https://<[URL Prefix>/service/user/attribute
  method POST
  eppn: gdhong@ex.com
  entity: gramtest
  api_secret: 0ef90fe
  attributes: [Identifiers of attributes]

Response:
{
  "code": 200
  "data": [
    {
      "name": "eduPersonEntitlement",
```

```
        "oid": "urn:oid:1.3.6.1.4.1.5923.1.1.1.7",
        "type": "M",
        "attribute_value": [
          "urn:mace:kafe.or.kr:gram:tncpje:test_grp",
        ]
      },
      {
        "name": "displayName",
        "oid": "urn:oid:2.16.840.1.113730.3.1.241",
        "type": "M",
        "attribute_value": ["Hong Gildong"]
      }
    ]
```


그림 6. 속성 질의 API의 예.
Fig. 6. Example of an attribute query API.

V. 검증 및 논의

이 장에서는 GRAM과 trustHUB의 구현 결과를 검증하고 개선 방향을 살펴본다. 스토리지 서비스에 사용자의 그룹 속성을 전달하는 시스템을 제안하는 본 연구의 목적이므로, 인증요청과 인증응답 메시지를 갈무리하여 사용자 속성이 정상적으로 추가나 제거되었는지 확인함으로써 구현 결과를 검증한다. SAML 메시지를 갈무리하기 위해 SAML-tracer를 사용했으며, API 요청 결과를 확인하기 위해 API 검증 도구인 Postman을 활용했다. SAML-tracer는 웹 브라우저에서 동작하는 플러그인 소프트웨어이다.

그림 7은 검증 환경과 검증 결과를 보여준다. 검증 환경은 실제 운영 중인 서비스들로 구성되어 있다. 스토리지 서비스(drive), 아이디제공자(coreen-idp), trustHUB (saml), 그리고 GRAM(gram) 서비스가 포함된다.

그룹 기반의 데이터 공유를 위한 속성으로, isMemberOf가 활용되도록 스토리지 서비스를 설정했다. 그림 6에서 확인할 수 있듯이 eduPersonEntitlement 속성값은 길이가 길어서 사용자가 여러 개의 속성값을 가지면 관리자가 그룹 정보를 확인하기 어려우므로, isMemberOf 속성을 사용했다. 스토리지 서비스는 isMemberOf 속성을 받은 후, 내부 규칙에 따라 사용자에게 데이터의 공유 권한을 부여한다. isMemberOf의 객체식별자(OID)는 1.3.6.1.4.1.5923.1.5.1.1이다.

이어서, 그림 7의 우측에서 볼 수 있듯이, GRAM에 스토리지 서비스의 V_s를 생성한 후 역할 그룹(G_s)으로 admin과 group_test를 지정했다. 마지막으로, 검증에 참여한 사용자를 두 역할 그룹에 할당했다.

그림 7의 왼쪽 아래에 있는 상자는 인증요청(AuthnRequest) 메시지를 trustHUB(saml)에서 갈무리한 내용을 보여준다. 사용자가 로그인을 시도하여 인증요청 메시지가 saml의 중계를 통해 스토리지 서비스(drive)에서 아이디제공자(coreen-idp)로 전달되었다. 갈무리된 내용 가운데 Issuer와 Destination은 각각 메시지의 발행지와 목적지를 의미한다. 인증요청 메시지가 coreen-idp에 도착하면 사용자는 자신의 자격 증명 정보(예, 사용자 아이디와 비밀번호)를 입력하여 인증을 받는다. 인증된 사용자의 속성정보가 saml에게 반환

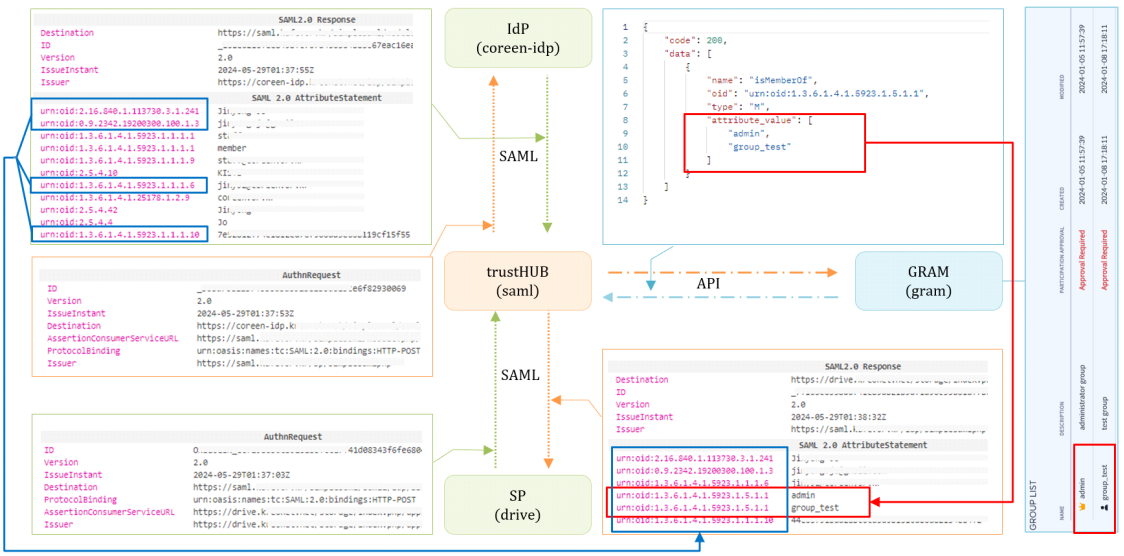

그림 7. 그룹 속성의 중계 과정 검증.
Fig. 7. Verification of the relay process of a group attribute.

된 것을 그림 7의 왼쪽 위 상자에서 확인할 수 있다. eduPersonPrincipalName(1.3.6.1.4.1.5923.1.1.1.6)은 사용자의 고유식별자이다.

그림 7의 오른쪽 위 상자는 trustHUB가 GRAM의 API를 호출해 얻은 그룹 정보를 보여준다. trustHUB와 GRAM 사이에는 브라우저의 개입 없이 직접 암호화 통신이 이루어지므로 메시지의 내용을 갈무리할 수 없다. 상자 안의 내용은 trustHUB에서 API 질의에 사용한 문자열을 Postman에서 동일하게 사용하여 얻은 결과이다. API 질의를 위한 킷값으로 eduPersonPrincipalName의 값을 사용했다. 해당 사용자의 isMemberOf 값으로 admin과 group_test가 정상적으로 반환되는 것을 확인할 수 있다.

마지막으로, 그림 7의 오른쪽 아래에서 확인할 수 있듯이, trustHUB는 coreen-idp로부터 얻은 속성과 gram으로부터 얻은 속성 중에서 drive에 전달하도록 설정한 속성만 전달했다. 그림 2의 속성 여과(Attribute filter) 모듈이 정상적으로 동작하고 있음을 보여준다. 결론적으로 그림 7의 우측 아래 상자에 나타난 내용을 통해 그룹 속성(예, isMemberOf)과 값(예, admin과 group_test)이 스토리지 서비스에 전달된 것으로 확인된다.

GRAM과 trustHUB의 개발 목표는 요구 기능의 간소화와 신속한 구현을 포함했으나 2개월 이상 서비스를 운영한 결과, 기능 확장이 필요한 것으로 확인되었다. 첫째, GRAM과 trustHUB 서비스의 이용 편의성을 높이기 위해 GRAM의 API 목록이 확대되어야 하며, 시스템과 사용자 간의 물리적 상호 작용이 최소화되어야 한다. 사용자가 GRAM을 직접 방문하지 않고도 로그인 과정에서 trustHUB가 그룹 가입 절차를 자동으로 처리할 수 있다면, 서비스의 이용 편의성을 높일 수 있다.

둘째, GRAM에서 사용하는 사용자 속성이 OIDC 인증규약이나 비표준 인증방식의 속성과도 호환되도록 시스템을 개선해야 한다. 현재 GRAM은 SAML의 속성 표기 규격을 채택하고 있다. 따라서 OIDC 인증규약을 사용하는 또는 비표준 인증방식을 사용하는 서비스 제공자와는 직접 연동되기는 어렵다. 예를 들어, SAML의 속성 uid는 OIDC의 sub와 대응될 수 있으나 시스템에서 uid만 관리한다면 sub를 키로 갖는 API 질의에 대해서 대응할 수 없다. GRAM 내부에서 별도의 속성 규격을 정의하고, SAML이나 OIDC 속성에 대응하는 변환 테이블을 활용한다면 이종 인증규약 간에 속성의 호환성을 확보할 수 있을 것으로 판단된다.

Ⅵ. 결 론

본 논문은 그룹 기반의 데이터 공유 환경에 적용하기 위해 개발한 GRAM과 trustHUB 시스템을 소개했다. 또한 구현된 시스템들을 신원연합 환경에서 실제 운영되고 있는 스토리지 서비스와 연계하여, 시스템의 구성 방식과 그룹 정보가 처리되는 과정을 보여줌으로써 그룹 기반의 데이터 공유를 실현하는 모범 사례를 제시했다. GRAM과 trustHUB의 개발과 활용이 표준 인증규약을 활용한 기관 간 데이터의 공유 환경을 조성함으로써, 국내 연구개발 사이버인프라의 고도화에 기여할 것으로 기대한다.

References

[1] M. Jirotka, C. P. Lee, and G. M. Olson, “Supporting scientific collaboration: Methods, tools and concepts,” *Comput. Supported Cooperative Work*, vol. 22, pp. 667-715, 2013. (https://doi.org/10.1007/s10606-012-9184-0)

[2] N. W. Jankowski, “Exploring e-science: An introduction,” *J. Computer-mediated Commun.*, vol. 12, no. 2, pp. 549-562, 2007. (https://doi.org/10.1111/j.1083-6101.2007.00337.x)

[3] B. Jacob, M. Brown, K. Fukui, and N. Trivedi, “*Introduction to grid computing*,” IBM redbooks, pp. 3-6, 2005.

[4] C. A. Stewart, S. Simms, B. Plale, M. Link, D. Y. Hancock, and G. C. Fox, “What is cyberinfrastructure,” in *Proc. 38th Annual ACM SIGUCCS Fall Conf.: Navigation and Discovery*, pp. 37-44, Oct. 2010. (https://doi.org/10.1145/1878335.1878347)

[5] T. Barton, et al., “Identity federation and attribute-based authorization through the globus toolkit, shibboleth, gridshib, and myproxy,” in *Proc. 5th Annual PKI R&D Wkshp.*, vol. 4, Apr. 2006.

[6] J. Basney, H. Flanagan, T. Fleury, J. Gaynor, S. Koranda, and B. Oshrin, “CILogon: Enabling federated identity and access management for scientific collaborations,” in *Proc. ISGC*, PoS(ISGC2019)031, 2019. (https://doi.org/10.22323/1.351.0031)

[7] Internet2, *COmanage Registry Deployment Guide*, Retrieved Jun. 5, 2024, from https://spaces.at.internet2.edu/display/COmanage/COmanage+Registry+Deployment+Guide.

[8] J. Jo, Y. Chae, and K. Kong, "Development of SAML-OIDC token translation system for web single-sign on," *J. KICS*, vol. 44, no. 10, pp. 1928-1938, Oct. 2019. (https://doi.org/10.7840/kics.2019.44.10.1928)

[9] T. István, et al., *HEXAA e-Science gateways with external attribute authority*, Retrieved Jun. 5, 2024, from https://geant3plus.archive.geant.net/Documents/HEXAA_AAI_Helsinki.pdf.

[10] T. Barton, P. Gietz, D. Kelsey, S. Koranda, H. Short, and U. Stevanovic, "Federated identity management for research," in *Proc. EPJ Web Conf.*, vol. 214, no. 03044, EDP Sciences, 2019. (https://doi.org/10.1051/epjconf/201921403044)

[11] A. Treloar, "Design and implementation of the australian national data service," *Int. J. Digital Curation*, vol. 4, no. 1, pp. 125-137, 2009. (https://doi.org/10.2218/ijdc.v4i1.83)

[12] J. Towns, et al., "XSEDE: accelerating scientific discovery," *Comput. in Sci. & Eng.*, vol. 16, no. 5, pp. 62-74, 2014. (https://doi.org/10.1109/MCSE.2014.80)

[13] J. Jo, H. Jang, J. Kong, and Y. Chae, "Federated IAM service of KAFE identity federation," *J. KICS*, vol. 43, no. 12, pp. 2200-2214, Dec. 2018. (https://doi.org/10.7840/kics.2018.43.12.2200)

[14] K. Cho, "Grid and e-Science in Korea," *Advances in Parallel Comput.*, vol. 18, pp. 464-482, 2009.

[15] *Internet2 Grouper*, Retrieved Jun. 5, 2024, from https://spaces.at.internet2.edu/display/Grouper/.

[16] *REFEDS MFA Profile*, Retrieved Jun. 5, 2024, from https://refeds.org/wp-content/uploads/2017/06/REFEDS-MFA-Profilev1.0.pdf.

[17] J. Jo, S. Kim, and B. Cho, "Development of TOTP verifier and proxied authenticator to enable strong authentication in identity federation," *J. KICS*, vol. 48, no. 10, pp. 1277-1288, Oct. 2023. (https://doi.org/10.7840/kics.2023.48.10.1277)

[18] R. Alfieri, et al., "VOMS, an authorization system for virtual organizations," in *Proc. Grid Comput.: First European Across Grids Conf.*, pp. 33-40, Feb. 2004. (https://doi.org/10.1007/978-3-540-24689-3_5)

[19] VOOT Specification, Retrieved Jun. 5, 2024, from https://github.com/frkosurf/voot-specification/blob/master/VOOT.md.

[20] J. Hughes, S. Cantor, J. Hodges, F. Hirsch, P. Mishra, R. Philpott, and E. Maler, *Profiles for the oasis security assertion markup language* (saml) v2.0. OASIS standard, Mar. 2005.

조 진 용 (Jinyong Jo)

2013년 : 광주과학기술원 정보통신공학과 박사
2003년~현재 : 한국과학기술정보연구원 책임연구원
2016년~현재 : 국제인증연합(eduGAIN) 운영그룹 위원
<관심분야> Trust and Identity, Networked applications and services
[ORCID:0000-0001-6830-3604]

장 민 석 (Min-seok Jang)

2011년 2월 : 한국과학기술원 전자및전자공학부 학사
2013년 8월 : 한국과학기술원 전자및전자공학부 석사
2013년 8월~2016년 12월 : (주)휴맥스 미주사업부
2016년 12월~현재 : 한국 과학기술정보연구원 과학기술연구망센터
<관심분야> 클라우드, 가상화, 네트워크 텔레메트리
[ORCID:0009-0008-1970-5760]

김 승 해 (Seung-Hae Kim)

2008년 : 전북대학교 정보보호공학 박사
1996년~현재 : 한국과학기술정보연구원 책임연구원
2021년~현재 : 한국과학기술정보연구원 연구망서비스팀 팀장

<관심분야> 라우팅 프로토콜 보안, 인증, 망관리, 정보보호

[ORCID:0000-0002-8403-7577]

조 부 승 (Buseung Cho)

2017년 : 성균관대학교 컴퓨터공학 박사
2005년~현재 : 한국과학기술정보연구원
2018년~현재 : 과학기술연합대학원대학교 데이터 및 HPC과학 부교수

<관심분야> 소프트웨어 정의 네트워크, 네트워크 관리

[ORCID:0000-0002-4661-5700]