재버를 위한 도메인 간 인증 메카니즘

최경선, 풍아평, 이이섭
금오공과대학교 컴퓨터공학과

e-mail: {choi540, fengyaping, eesub}@kumoh.ac.kr

Inter-Domain Authentication Mechanism for jabber

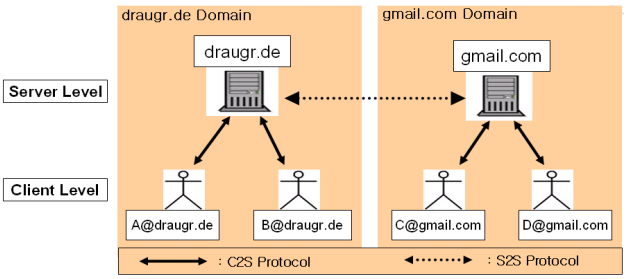
Kyung-Sun Choi, Ya-ping Feng, Lee-Sub Lee
Dept of Computer Engineering, Kumoh University

요 약

IDA(Inter-Domain Authentication)는 한 영역의 클라이언트가 다른 영역의 서버에 접근이 가능하게 하는 인증 메카니즘이다. 기존의 커버러스 인증 메카니즘에는 IRA(Inter-Realm Authentication)를 제공하고 있지만 모든 인증방식이 커버러스로 구현된 경우에만 가능하다. 현재 재버의 표준으로는 커버러스 이외에 다른 인증 메카니즘이 병존할 수 있기 때문에 커버러스의 IRA로는 지원이 불가능하다. 따라서 본 연구에서는 재버의 환경에서 다양한 인증 메카니즘이 존재하는 경우에도 적용할 수 있는 IDA를 제안한다. 이 메카니즘을 사용함으로써 중복저장과 동기화에 대한 문제를 해결하여 분산 응용프로그램을 용이하게 설계 구축 할 수 있다.

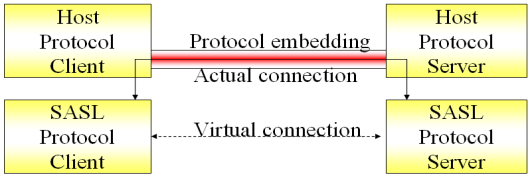
1. 서론

재버(Jabber)는 인터넷 상의 임의의 두지점 사이의 메시지와 프레즌스의 실시간 교환을 위한 XML 기반의 개방형 프로토콜이다. AOL이나 MSN과 같은 다른 IM(instant message) 네트워크와 함께 연동 가능한 인스턴트 메시징 플랫폼이다. 재버는 DNS와 URI 기반한 주소체계를 사용하여 전자우편에서 사용되는 것과 똑 같은 형태의 주소를 사용하고 있다. 재버는 XML 기술을 사용하여 어떤 구조의 데이터도 표현할 수 있기 때문에 확장성이 높아 광범위한 분야의 분산 응용프로그램 개발이 가능하다. 따라서 최근 재버를 기반으로 한 xDash(EAI), 재버 메일 컴포넌트(전자우편)등과 같은 분산 응용프로그램을 개발하는 프로젝트가 진행 중이다. [2,3] 재버를 기반으로 분산 응용프로그램을 재버에 구축하는 경우의 문제점을 살펴보기 위하여 먼저 재버의 구조를 분석하여 보자. 아래의 <그림1>에서 알 수 있듯이 전자우편 시스템과 유사한 구조를 가지며 XML 기반의 메시지를 사용한다는 것이다. 첫째로 재버 환경은 하나의 재버 서버와 이에 속한 클라이언트들로 이루어진 도메인들로 구성되어 있다. 둘째로 각 도메인은 전통적인 C2S (Client-to-Server) 구조로 구성되어 있다. 셋째로 하나의 클라이언트는 소속된 서버와 C2S 프로토콜로 연동된다. 넷째로 서버 간에는 S2S(Server-to-Server) 프로토콜로 연결된다. 기존의 구조에서는 특정 클라이언트가 다른 도메인의 서버와 연결이 불가능하다. [1] 재버는 인터넷상에서 누구나 서버를 구축하여 참여할 수 있으므로 보안 문제가 가장 중요하다. 보안의 가장 기본적인 기능으로서 C2S 인증과 S2S 인증을 지원하고 있다. C2S 인증은 다양한 인증기법을 모두 지원 가능하다. C2S 인증의 종류로는 익명인증, 간단한 원문인증, 요약 인증, 제로-날리지 인증이 있다. S2S 인증은 다이얼백 인증을 지원 하고 있다. 최근에는 SASL (Simple Authentication and Security Layer) 이 재버 인증 표준에 추가 되었다.



<그림1> 재버 구조

추가된 SASL에 대한 설명은 <그림2>로 나타내고 있다. 그림에서 SASL의 프로토콜 삽입(Protocol embeding)이란 기존 연결 기반의 프로토콜에 필요한 메소드를 추가하여 가상연결을 통해 인증서비스를 제공하는 기술이다.



<그림 2> SASL

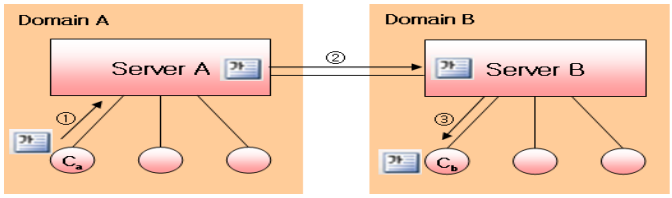
SASL은 플러거블 인증(Pluggable authentication)을 제공하고 기존 프로토콜에 인증 기능을 추가 시키는 기능을 제공하고 있다. 또한 이미 검증된 커버러스, S-Key, External, GSSAPI 인증 메카니즘을 제공하고 있으며 SASL은 재버 인증 기능을 추가하기 위한 최선의 방법으로 인식되어 최근 재버 표준에 추가 되었다. [4] 본 논문에서 새로운 IDA인증 메카니즘을 제안하는 이유는 재버의 특성에서도 알 수 있듯이 현재 인터넷 상에는 수 많은 도메인 간의 다양한 인증메카니즘을 존재하고 있다. 예를 들어 보자. A도메인은 현재 커버로스 환경으로 되어 있고 B 도메인은 현재 S-Key 환경으로 되어 있다. 이때 A도메인의 서버나 클라이언트가 B도메인에 접근하고자 할때 A도메인은 B 도메인과 같은 환경 일때만 접근할 수 있다. 즉 도메인 B는 커버러스와 같은 환경이 아니므로 도메인 A는 도메인 B에 접속할수 없게 된다. 그러므로 본 논문에서 이와같은 서로 다른 메카니즘 환경에서도 서로 호환할수 있는 IDA를 제공하고자 한다. 결론적으로 SASL은 재버 환경에서 다양한 인증메카니즘을 동시에 지원해야 함으로 본 논문에서는 IDA를 통해 다양한 인증 메카니즘이 존재하는 경우에도 사용할 수 있는 IDA를 제안하고 있다. 본 논문의 2장에서는 IDA의 필요성에 대하여 파일시스템과 워크플로우의 시스템의 문제점을 설명하고 IDA 적용시스템과의 비교를 통해 차이점을 기술한다. 3장에서는 IDA를 제시하고 수행과정에 대해서 기술한다. 4장의 관련 연구에서는 보안의 가장 중요한 요소인 인증 부분에서 XML전자서명에 대해 살펴보고 XML 전자서명의 인증방식인 PKI 기반 인증 방식과 IDA 기반 인증 방식의 비교를 통해 인증방식의 특징과 문제점들을 살펴 본다. 그리고 커버러스 환경에 대한 IDA 시스템 환경과의 비교를 하고 5장에서 결론을 맺는다.

2. IDA의 필요성

기존의 재버 인증 모델에서는 한 영역의 클라이언트가 다른 도메인 영역의 서버에 직접 접근하는 기능이 제공되지 않으므로 분산 응용프로그램 개발시 다음과 같은 데이터 중복과 동기화 문제가 발생된다. 재버 서버시스템의 문제점을 분석하고 이를 통해 얻어진 사항을 기반으로 IDA 적용시스템과의 비교를 통해 차이점을 기술한다. 아래의 예를 통해 IDA의 필요성을 살펴보도록 하자

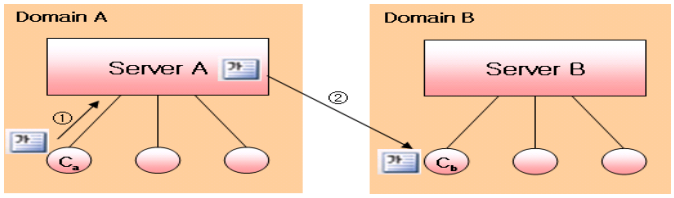
2.1 파일전송 분산 응용프로그램의 예

현재의 일반적인 재버 서버 시스템의 파일 전송시에 도메인 A클라이언트가 B서버의 클라이언트에게 정보를 전송하고자 할때 기존의 파일 전송 시스템의 구조를 살펴 보면 아래의 <그림3>과 같이 나타낼 수 있다. 기존의 시스템에 대한 설명하기 위해서 아래 그림은 파일전송 응용프로그램 수행시의 과정을 보여 준다. 여기에서 C_a, C_b는 클라이언트를 의미하며 A와 B는 서버를 의미한다.



<그림 3> 일반적인 재버 서버 시스템

위 그림에서 알 수 있듯이 첫 번째 단계로 C_a가 A에 파일을 전송 한다. 두 번째 단계로 A가 B에게 파일을 복사한다. 세 번째 단계로 B는 C_b에게 파일 전송 한다. 이 과정에서 알 수 있듯이 데이터 중복의 문제와 서버 동기화에 대한 문제가 발생하고 있다. 파일전송 분산 응용프로그램의 예의 대부분의 경우는 여러 도메인에 걸쳐 작업이 발생하기 때문에 심각한 문제가 되고 있다. 그렇다면 기존의 시스템의 문제를 해결하기 위하여 본 논문에서 제안한 IDA를 재버 시스템에 적용하게 되면 <그림4> 와 같이 나타낼 수 있다. IDA를 적용하여 분산 환경에서 발생할 수 있는 해결책을 제시하기 위해 본 논문에서는 현재 재버 시스템에 IDA를 적용한 환경에서 기존의 환경에서의 한계를 벗어나 다음과 같은 이점을 제공하여 준다. 이를 위해서 IDA를 적용한 시스템의 수행 절차를 그림으로 표현해 보면 다음과 같다.



<그림4> IDA 적용 재버 서버 시스템

위의 그림에서 알 수 있듯이 첫 번째 단계로 C_a가 A로 파일을 전송 한다. A는 이를 확인하고 B와의 상호인증을 맺음으로 인해서 A가 C_b에게 직접 파일 전송 할 수 있다. 위의 과정에서 알 수 있듯이 IDA를 적용한 시스템의 경우 데이터 중복의 방지를 통해 전송 비용을 절감 할 수 있다. 재버의 환경은 이메일과 같이 공개 되어져 있기 때문에 매우 많은 도메인들이 존재 하고 있다. 각각의 도메인간의 상호인증을 미리 가질 수 없고 임의의 클라이언트가 다른 영역 도메인에 속한 클라이언트와의 협업이 가능하게 위해서는 필요시에 협업을 위해 서버간 상호인증을 맺는다. IDA에서는 서버간의 상호인증을 통해서 다른 영역의 서버가 다른영역의 클라이언트에 직접 접근 할 수 있다. 위의 그림을 통한 일반적인 재버 시스템과 IDA 적용 재버 서버시스템의 전송량과 요구되는 예상 저장 공간에 대한 계산 공식을 도식하여 보면 <표1>과 같다.

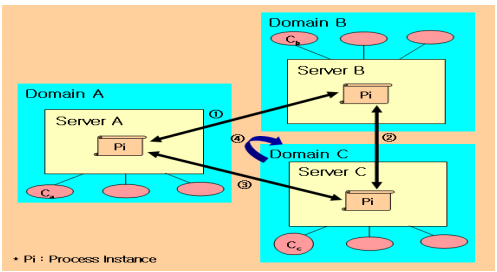
표 1. 적용 시스템별 계산공식

적용 시스템	전송량	요구되는 데이터 공간
기존 방식	(1+2*N)*S)	(1+N)*S
IDA	(1+N)*S	S

N : the number of distinct destination domains
S : the size of transmitted file

2.2. 분산 워크플로우의 예

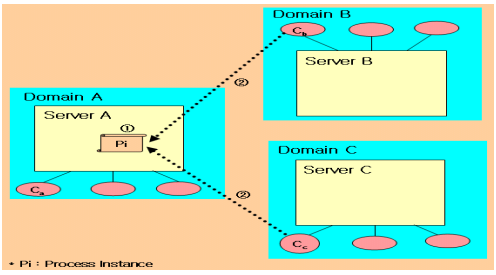
현재의 일반적인 재버 서버 시스템에 기반한 워크플로우 [10] 시스템을 구현한다고 가정해 보면 아래 <그림5>와 같다. 이 그림은 가정은 다음과 같다. 도메인 A의 서버A는 도메인 B의 서버 B와 도메인 C의 서버C에 대하여 협업을 위한 워크플로우 Pi 데이터를 교환하고자 한다. 현재의 분산 워크플로우 시스템은 어떻게 동작하고 있는지 그림을 통해 현재의 시스템 수행절차를 나타내였다. 여기에서 C_a, C_b, C_c는 클라이언트를 의미하며 A와 B 그리고 C는 서버를 의미한다. 또한 Pi는 프로세스 인스턴트를 나타내고 있다.



<그림5> 기존 워크 플로우 시스템

위 그림에서 알 수 있듯이 첫 번째 단계로 C_a가 C_b, C_c가 참여하는 Pi(process Instance)를 생성한다. 두 번째 단계로 C_b와 C_c는

A에 저장된 Pi를 직접접근 할 수 없기 때문에 B와 C에 Pi 복사가 이루어진다. 세 번째 단계로 각 클라이언트는 자신이 소속된 서버에 중복된 Pi Data를 사용한다. 네 번째 단계로 각 클라이언트가 중복된 데이터를 독립적으로 갱신 할 수 있으므로 동기화가 필요하다. 현재의 시스템의 문제점은 다음과 같다. 기존 워크 플로우 시스템의 경우 파일 전송의 경우와 같이 전송량의 문제와 중복되는 데이터 저장 공간 낭비문제가 발생한다. 또한 동기화의 문제로 n개의 Pi가 중복되어 각 클라이언트가 동시변경 가능하므로 동기화 문제가 발생하게 된다. 그러므로 현재 시스템을 보다 자세히 살펴 보면 보안(Security)문제가 발생할 수 있다. 위 그림에서 알수 있듯이 보안에 민감한 Pi 정보가 모든 서버에 중복 되므로 보안 유지의 어려움이 발생한다. 위의 문제점 해결을 위해 IDA 적용한 워크플로우 시스템은 <그림6>과 같다. 아래의 그림은 IDA 적용 워크플로우 시스템 수행 절차 순서는 나타내고 있다.

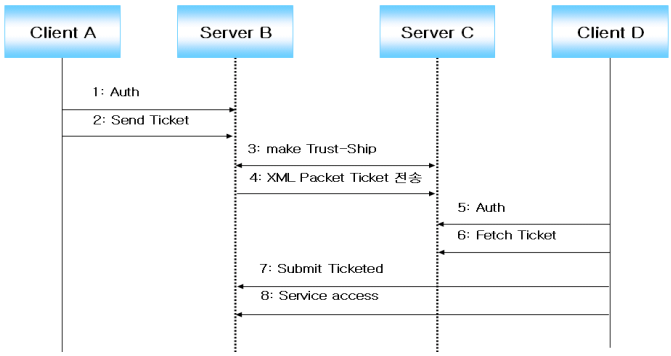


<그림6> IDA 적용 워크플로우 시스템

첫 번째 단계로 C_a가 C_b, C_c가 참여하는 Pi를 생성한다. 두 번째 단계로 B의 경우 C_b, C_c가 A에 접근 할 수 있기 때문에 중복할 필요가 없다. 세 번째 단계로 각 클라이언트는 A의 Pi 정보에 접근하여 작업을 수행한다. 위의 그림을 통해 알 수 있듯이 IDA적용 워크플로우 시스템의 예를 통해 기존 워크플로우 시스템에 대한 문제점을 해결할 수 있다. 그렇다면 파일전송 분산응용프로그램 시스템과 워크플로우 시스템의 어떤 문제를 해결 할 수 있는지를 살펴 보도록 하자. 기존의 파일전송시스템은 동일한 전송량 데이터의 감소와 데이터 중복의 문제를 해결할 수 있다. 기존의 워크플로우 시스템은 해당서버에 직접 접근함으로써 동기화 문제를 해결할 수 있다. 또한 A가 Pi데이터를 관리함으로써 보안에 대해 안전한 관리를 할 수 있다. 따라서 IDA를 사용하면 훨씬 용이한 분산 응용 프로그램을 개발 가능하게 하며 워크플로우 시스템의 경우 프로세스 공유에 있어 발생하는 문제점을 해결 할 수 있다.

3. IDA

IDA(Inter-Domain Authentication)는 한 영역의 클라이언트가 다른 영역의 서버에 접근이 가능하게 하는 인증 메카니즘이다. 위의 예를 통해 IDA 적용 시스템에 대한 문제점 해결 방안을 알 수 있다. 본 절에서는 IDA 인정과정에 대해 살펴 보도록 하자. IDA 인증 과정은 <그림7>과 같다. 그림에서 클라이언트 A가 서버 C의 클라이언트 D와의 협업을 수행하기 위한 과정을 개념적으로 표현하고 있다. 그러면, 수행되는 절차를 살펴보도록 하자. 첫 번째 단계로 A가 C2S인증을 이용하여 B로부터 인증을 받게 된다. 두 번째 단계로 A는 D@C와의 협업을 위하여 티켓(Ticket)을 사용하여 재버의 C2S 프로토콜로 B에게 전송한다. 티켓에는 1회의 서비스를 사용할 수 있는 권한이 저장되어 있다. 세 번째 단계로 B는 D가 속한 C와 SASL을 이용하여 상호인증을 수행한다. 재버 환경은 전자우편과 같이 공개 되어져 있기 때문에 매우 많은 도메인이 존재하여 임의의 다른 도메인간의 상호인증을 미리 가질 수 없다. 따라서 임의의 클라이언트가 임의의 다른 도메인에 속한 클라이언트와 협업이 가능하도록 필요시에 상호인증을 맺는다. 네 번째 단계로 B는 C에게 S2S 프로토콜을 사용하여 티켓을 전송한다. 다섯 번째 단계로 D가 C2S인증을 이용하여 C로부터 인증을 받게 된다. 여섯 번째 단계로 D는 C로부터 티켓을 확인하게 된다. 일곱 번째 단계로 D는 서비스 접근 티켓을 B에 전송하여 인증을 받는다. 여덟 번째 단계로 서비스를 직접 접근한다. IDA는 이와 같이 시스템에 적용할 수 있다.



<그림7> IDA 인증과정의 개요

4. 관련연구

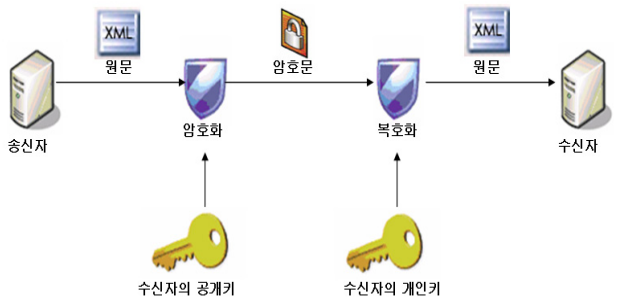
본 절의 4.1절에서는 XML전자서명 대해 살펴 보고자 한다. 이를 통해서 기존의 PKI기반의 XML 전자서명 인증방식과 IDA 인증방식을 비교하여 4.2절에서 설명하고자 한다. 4.3절에서는 기존의 커버러스 환경과 IDA의 환경을 비교하여 기존의 커버러스 환경에서 어떤 문제점이 있고 이 문제점에 대한 해결방안을 모색하고자 한다.

4.1 XML 전자서명

XML 전자서명은 W3C의 XML-Signature Working Group에서 제정하였다. XML 문서에 대한 전자서명을 할 수 있는 규칙과 구문처리를 명시하고 PKI(Public Key Infrastructure) 구조로 되어 있다. 전자서명이라는 것은 사용자의 신원이나 문서의 변조 여부를 확인할 수 있는 방법으로 발신인의 신원을 증명할 수 있는 개인의 인감도장이나 싸인 같이 실제 작성자를 확인 할수 있는 방법이다. 현재의 인터넷을 통한 문서 전달에는 문서의 실제 작성자를 확인 할 수 있는 방법이 필요하다. 인터넷 상에서 문서를 작성한 송신자와 문서를 읽는 수신자를 확인하기 위한 방법으로 무결성과 인증 서비스를 제공하는 것이 전자 서명이다. 전자서명은 해쉬 알고리즘을 통해 문서의 변조 여부를 판단할 수 있다. 해쉬 함수를 사용하여 문서의 일부분을 변조하여도 해쉬 함수의 축약문은 달라짐으로써 문서의 변조 여부를 알 수 있다. 또한 해쉬 함수로 만들어진 축약문은 개인키를 가진 사용자만이 볼 수 있도록 암호화 되어져 있기 때문에 개인키를 보유한 사용자가 암호를 복호화 할 수 있다. 이를 위해서 전자 서명에서는 PKI 기반의 공개키 기반 구조를 사용하고 있다.

4.2 PKI 기반 인증 방식과 IDA 기반 인증 방식 비교

PKI 기반 인증 방식은 XML 전자서명에 이용되고 있다. <그림9>에서 PKI의 기본 구조를 살펴 보도록 하자. 이 그림에서 수신자의 공개키와 비밀키는 인증기관으로부터 제공받은 것이다. 이 그림을 통해 알 수 있듯이 PKI 인증 방식은 공개키 값의 안전하고 효율적인 전송을 위해 인증서를 발행, 획득, 조회, 검증 등을 수행하는 인증서 관리 기반 구조를 말한다. 공개키 암호화 알고리즘은 문서를 실제 사용하게 될 수신자의 공개키로 암호화 하게 되고 수신자는 자신의 비밀키로 축약문을 복호화 해서 문서의 변조 여부를 알 수 있다. PKI 인증 방식은 전자서명 애플리케이션에서 무결성, 부인봉쇄, 인증등의 보안 서비스를 효율적이고 안정적으로 제공하는 것을 목적으로 하고 있다.



<그림9> PKI 인증 방식

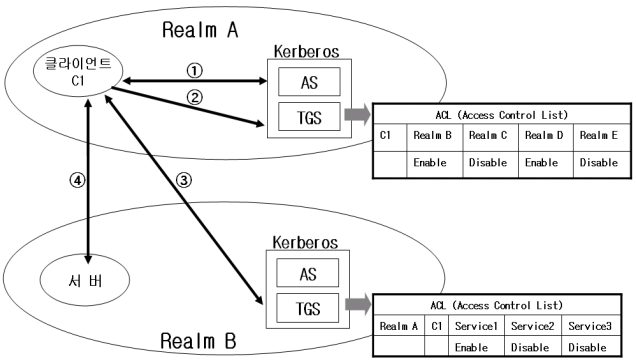
현재 PKI는 비대칭키를 사용하고 있다. 보안성 측면에서 보면 개인키와 공용키를 사용함으로써 강력한 암호화 기능을 제공하고 있다. 반면 IDA는 대칭키를 사용하고 있으므로 특정 클라이언트만이 해독할 수 있는 비밀키 방식의 암호화를 제공하고 있다. PKI의 암호화 속도는 인증기관에 대해 공개키를 제공 받고 암호화 하고 복호화 하는 과정으로 인해 암호화의 시간이 오래 걸리며 인증기관의 설치로 인한 비용이 증가되며 별도의 인증기관에서 상호인증을 위한 비대칭키를 사용함으로써 중앙집중식의 문제점인 Single-Point-Failure의 문제가 발생하기 때문에 안정성이 문제가 된다. 반면 IDA의 암호화 속도는 PKI에 비해 상대적으로 빠른 암호화를 제공하고 있다. IDA는 특정 클라이언트에 대한 신속한 접근을 할 수 있고 인증기관의 미설치로 인한 비용절감과 Single-Point-Failure 관리가 필요 없다. 현재의 재버 환경은 인터넷 상에서 수 많은 도메인 서버 또는 클리이언트 간의 데이터 교환을 필요로 하고 있다. 만약 IDA에 강력한 암호화를 위해 PKI를 적용하였을 경우 인증기관을 통해 강력한 암호화 기능을 제공할 수 있지만 PKI 기반으로 IDA를 제공할 경우에 인증기관의 관리에 대한 비용이 증가 하며 Single-Point-Failure 문제가 발생하게 된다. 또한 도메인 서버 및 클라이언트는 인증기관의 처리로 인해 인증기관 부하 발생 및 많은 저장 공간이 필요하다. 수 많은 도메인 서버 또는 클라이언트 간의 여러 다른 전자서명을 가지는 경우에 수 많은 전자서명의 생성, 배분, 보관, 복구 등의 부담으로 인증기관의 운영이 불가능하다. 그러므로 본 논문에서는 IDA에 대칭키 방식의 암호화를 적용 하였다.

4.3 커버러스 환경 및 IDA 환경의 비교

현재 잘 알려진 커버러스 환경의 IRA를 <그림9>와 같이 나타내 보았다. 다음은 커버러스의 IRA 수행과정을 나타내고 있다.

① 클라이언트가 커버러스에 접속 한다.
② 클라이언트가 원격 영역에 접속을 위한 티켓을 확보 한다.
③ 클라이언트가 원격 커버러스로 부터 원격 영역의 서비스 사용티켓을 확보 한다.
④ 티켓을 사용하여 서비스에 접속 한다.

이에 대하여 어떤 문제점이 있는지를 분석해 보기로 하자. 첫 번째 지역 영역과 원격 영역이 모두 커버러스로 구성되어야 하는 문제점이 있다. 그러나 재버의 경우 다양한 메카니즘을 모두 포함하고 있으므로 적용이 어렵다. 두 번째 ②의 경우에서 영역의 개수가 매우 많기 때문에 모든 도메인이 커버러스 만을 인증 메카니즘으로 사용한다고 가정을 해도 지역 커버러스가 모든 클라이언트에 대해서 원격 영역의 접근 권한 리스트를 유지하기가 어렵다.[8,9] 반면 IDA를 사용하게 되면 첫 번째 문제는 다양한 메카니즘을 사용함으로 해결 할 수 있고 두 번째 문제는 필요시에 상호인증을 맺음으로 이 문제를 해결 할 수 있다.



<그림10> 커버러스의 IRA

5. 결론

본 논문에서는 현재의 재버 환경에서 문제점을 살펴 보고 이를 통해 발생하고 있는 문제점 해결을 위하여 분산 응용프로그램과 워크플로우 시스템에 대한 IDA를 적용하여 보았다. 이를 통해 재버 환경에서 한 영역의 클라이언트가 다른 영역 도메인의 서버에 직접 접근 할 수 있는 인증 모델을 제안하였다. 이 모델을 적용함으로써 분산 응용프로그램 개발에 가장 큰 장애가 되는 중복과 동기화 문제를 용이하게 해결 할 수 있다. 기존의 PKI 기반 인증구조와 IDA 기반 인증구조의 비교를 통해서 현재 재버 시스템의 인증에 대한 보안상의 문제점이 상존하고 있음을 알 수 있다. 제안된 모델은 재버와 같이 다양한 인증 메카니즘이 병존하는 환경에 적합하다. 이는 기존 커버러스의 IRA에서는 불가능한 기능이다.

향후의 과제로는 IDA의 부인방지에 대한 문제점 해결방안에 대한 연구가 있을 것이라 판단 된다. 또한 PKI의 장점을 살리고 인증기관의 지원없이 IDA 기반에 적용할 수 있는 시스템에 대한 연구를 계속 진행하고자 한다.

참고문헌

[1] I. Shigeoka, "Instant Messaging in Java" MANNING, 2002
[2] "http://www.jabber.org", Jabber Software Foundation
[3] "http://www.jabberstudio.org", development hub for the jabber community
[4] "http://www.faqs.org/rfcs/rfc2222.html" RFC2222 Simple Authentication and Security Layer (SASL)
[5] J. Mitchell, "Logic for Computer Security Protocols", Stanford University
[6] 두소영, "신뢰성이 있는 사용자 인증 시스템과 안정성 분석", 2003.
[7] M. Burrows, M. Abadi and R. Needham. "A Logic of Authentication." Report 39 Digital Systems Research Center. 1989
[8] M. Steven, "Limitations of the Kerberos Authentication System" AT&T Bell Laboratories
[9] J. T. Kohl, B. C. Neuman, Y. T. Theodore, "The Evolution of the Kerberos Authentication Service" Digital Equipment Corporation
[10] R. Allen, "Workflow: An Introduction" Open Image Systems Inc., United Kingdom Chair, WfMC External Relations Committee