웹기반 그리드 인증서 및 키관리 시스템의 개발

김상완*, 박형우**, 이상산***
KISTI 슈퍼컴퓨팅센터
e-mail: {sangwan*, hwpark**, sslee***}@hpcnet.ne.kr

A Development of Web-based Grid Certificate and Key Management System

Sang-Wan Kim
KISTI Supercomputing Center

현재 그리드 컴퓨팅 툴킷으로 많이 사용되고 있는 Globus (www.globus.org)는 PKI (Public Key Infrastructure) 기반의 인증방식을 이용하고 있다. 그리드 컴퓨팅은 많은 컴퓨터, 프로그램, 그리고 사람이 동시에 참여하는 컴퓨팅 기술이다. 따라서 컴퓨터와 사용자에 대한 인증서와 키를 체계적이고 사용자 측면에서 효율적으로 관리할 수 있는 방안이 요구된다. 본 연구에서는 사용자들이 자체적으로 인증기관을 운영하는데 있어 편리하게 인증서를 관리할 수 있는 도구를 개발하였다. 본 연구에서 개발된 웹기반 인증서 관리도구를 이용하여, Globus를 사용하기를 원하는 기관에서는 보다 체계적이고, 자동화된 방법으로 인증서와 키를 관리할 수 있게 된다.

1. 서론

그리드(Grid)[1]란 다양한 기관에 지역적으로 분산되어 있는 다양한 종류의 컴퓨팅 자원을 네트웍을 이용하여 보다 효율적으로 사용하는 컴퓨팅 방식이다. 그리드는 메타컴퓨팅(metacomputing)이라는 이름으로 1980년대 후반에 생겨나기 시작하였으나, 최근에 컴퓨터 네트워크 기술의 발전으로 인해 다시 관심이 집중되고 있는 연구분야이다. Globus[2]는 현재 개발되고 있는 그리드 서비스를 제공하는 미들웨어로 가장 많이 이용되고 있는 것으로써, 미국의 ANL (Argonne National Laboratory) 에서 개발이 진행중이다.

Globus에서는 계산 자원의 서비스 요청에 대한 인증을 수행하는 과정에서 PKI (Public Key Infrastructure)[5]방식을 이용하고 있다. PKI기반의 인증방식에서는 모든 인증 객체에 대해서 인증서와 키가 부여되는데, 이것을 체계적으로 발급, 관리, 갱신, 폐기하는 것이 중요하다. Globus에서도 기계와 사용자에 대한 인증서와 키를 발급하여 사용하도록 하고 있는데, 그 과정에서 인증요청서를 인증기관에 이메일로 보내고, 인증서를 이메일로 받아 설치하는 과정을 요구하고 있다. 본 연구에서는 Globus의 설치 과정에서 인증서의 발급 및 관리를 간단하게 자동화해 주는 웹기반의 도구를 개발하였다. 본 논문의 구성은 다음과 같다. 제2장에서는 기존 Globus에서 인증서의 종류와 발급과정을 설명하고, 제3장에서는 본 연구에서 구현한 인증서 관리 도구의 기능과 사용방법을 설명한다. 제4장에서는 앞으로의 개발 방향과 계획을 설명하고, 마지막으로 제5장에서 결론을 맺는다.

2. 현재 Globus에서 인증서 종류와 인증절차

Globus에서 사용되는 인증서의 종류는 4 가지로 나누어 볼 수 있다. 첫 번째는 Globus가 설치되어 있는 기계(또는 계산 자원)에 대한 인증서이다. Globus에서 어떤 자원에 대한 접근은 맨 처음 게이트키퍼(gatekeeper)라고 불리는 호스트 컴퓨터에 대한 인증으로부터 시작된다. 이 게이트키퍼 인증서는 CA가 게이트키퍼에게 발급하며, CN (Common Name)으로 게이트키퍼의 호스트명을 사용하고 있다. 그림1은 게이트키퍼인증서의 예를 보여준다.

두 번째는 Globus 사용자의 인증서이다. 이 인증서는 CA가 Globus 사용자에게 발급하며, OU (Organizational Unit)와, CN으로 사용자의 이름에 대한 정보를 포함하고 있다. 그림2는 사용자 인증서의 예를 보여준다.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 22 (0x16)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=KR, O=Globus, CN=KISTI Certificate
Authority
        Validity
            Not Before: Feb 17 13:04:13 2002 GMT
            Not After : Feb 17 13:04:13 2003 GMT
        Subject:          O=Grid,          O=Globus,
CN=sdd110.hpcnet.ne.kr
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
...(생략)...
    Signature Algorithm: md5WithRSAEncryption
...(생략)...
```

<그림1> Globus 게이트키퍼 인증서

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 23 (0x17)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=KR, O=Globus, CN=KISTI Certificate
Authority
        Validity
            Not Before: Feb 17 13:08:55 2002 GMT
            Not After : Feb 17 13:08:55 2003 GMT
        Subject: O=Grid, O=Globus, OU=hpcnet.ne.kr,
CN=globus
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
...(생략)...
    Signature Algorithm: md5WithRSAEncryption
...(생략)...
```

<그림2> Globus 사용자 인증서

세 번째는 사용자 프록시(proxy) 인증서이다. 앞의 두 가지 인증서가 기본적으로 1년이라는 비교적 긴 유효기간을 가지고 있는데 반해 사용자 프록시 인증서는 사용자가 지정한 시간 동안만(기본 12시간) 효력을 가진다. 사용자는 Globus 서비스를 이용하기 전에 Globus의 grid-proxy-init라는 명령을 사용하여 자신의 프록시 인증서를 생성시켜야 한다. 그림3은 사용자 프록시 인증서의 예를 보여주는데, 발행자(issuer)가 Globus 사용자로 표시되어 있음을 알 수 있다.

마지막으로 Globus에서 사용되는 네 번째 인증서는 프로세스 또는 자원(resource) 인증서로써, 외부의 요청에 의해서 자원이 할당되어 프로세스가 실행되는 동안만 생성되어 사용되는 인증서이다. 그림4는 프로세스 인증서의 예를 보여준다. 이 인증서의 발행자는 사용자 프록시임을 알 수 있다.

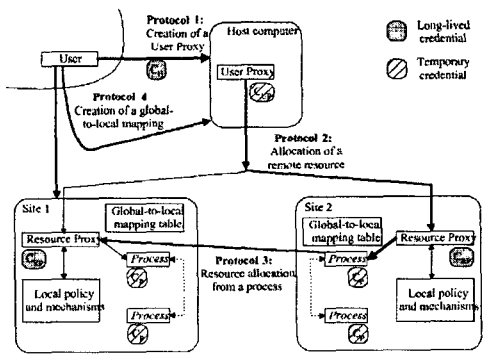
그림5는 Globus에서 인증 과정을 설명하고 있는다. 사용자는 자신의 사용자 인증서(C_U)를 이용하여 사용자 프록시 인증서(C_{UP})를 생성시킴으로써, 자신이 개입되지 않고도, 그리드 상에 있는 어떤 자원도 이용할 수 있도록 한다. 사용자가 원격에 있는 자원에 대한 사용 요구를 하게 되면, 원격지의 게이트키퍼에 있는 인증서(C_{RP})를 이용하여 상호인증(mutual authentication)이 수행되고, 프로세서가 성공적으로 생성되면, 프로세스 인증서(C_P)가 만들어진다. 이 것은 프로세서가 또 다른 원격지에 있는 자원에 접근할 때 자신을 상대방에게 인증시키기 위한 목적으로 사용된다.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 23 (0x17)
        Signature Algorithm: md5WithRSAEncryption
        Issuer:  O=Grid,  O=Globus,  OU=hpcnet.ne.kr,
CN=globus
        Validity
            Not Before: Mar  9 07:25:53 2002 GMT
            Not After : Mar  9 19:30:53 2002 GMT
        Subject:  O=Grid,  O=Globus,  OU=hpcnet.ne.kr,
CN=globus, CN=proxy
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (512 bit)
...(생략)...
    Signature Algorithm: md5WithRSAEncryption
...(생략)...
```

<그림3> Globus 사용자 프록시 인증서

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 23 (0x17)
        Signature Algorithm: md5WithRSAEncryption
        Issuer:  O=Grid,  O=Globus,  OU=hpcnet.ne.kr,
CN=globus, CN=proxy
        Validity
            Not Before: Mar  9 07:27:11 2002 GMT
            Not After : Mar  9 19:30:53 2002 GMT
        Subject:  O=Grid,  O=Globus,  OU=hpcnet.ne.kr,
CN=globus, CN=proxy, CN=limited proxy
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (512 bit)
...(생략)...
    Signature Algorithm: md5WithRSAEncryption
...(생략)...
```

<그림4> Globus 프로세스 인증서



<그림5> Globus Security Architecture
([3]에서 인용)

Globus에서 CA가 발급해 주어야 하는 인증서는 사용자 인증서와 게이트키퍼 인증서 두 종류이다. Globus에서는 관례상 모든 인증대상의 DN(Distinguished Name)은 O=Grid, O=Globus로 시작하도록 하고 있다. Globus에서 게이트키퍼의 인증서와 비밀키는 기본적으로 $(BLOBUS_DEPLOY)/etc/ 에 사용자 인증서는 $(HOME)/.globus/ 에 설치된다. 이 두 가지 인증서를 설치하기 위해서 관리자와 사용자는 인증요청서를 이메일주소(ca@globus.org)로 보내고 인증서를 받도록 하고 있다.

3. 웹을 통한 인증서 발급, 관리 도구

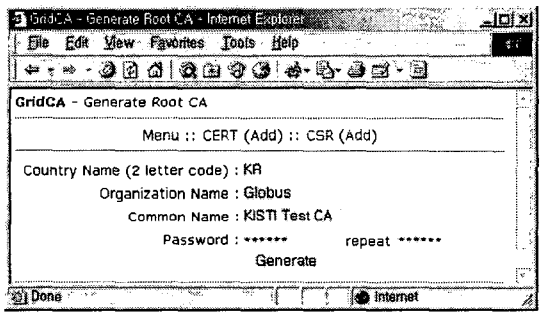
이상에서 설명한 Globus에서 인증서 발급과 설치 과정은 Globus를 설치하여 시험하여 보려는 사람들에게 상당히 번거로운 과정일 뿐 아니라, CA의 도움이 없이는 인증서를 설치할 수 없기 때문에 Globus의 시험 운용에 제약이 따르지 않을 수 없다. 본 연구에서는 Globus의 인증서 발급과 관리를 자동화 할 수 있는 웹기반의 도구를 개발하였다.

Globus에서 GSI(Grid Security Infrastructure)는 Eric Young의 OpenSSL[4] 라이브러리를 이용하여 구현되었기 때문에, OpenSSL에서 제공하는 인증서 관리 명령어를 이용하면 손쉽게 인증서를 생성시킬 수 있다. CA의 인증서, 게이트키퍼 인증서, 그리고 사용자 인증서는 DN의 구조가 모두 다른데, DN의 구조는 OpenSSL의 설정화일에서 정의되므로, 설정화일을 바꾸어서 인증서를 생성시키면 서로 다른 구조를 가진 DN의 구조를 가진 인증서를 만들어 낼 수 있다. 그림6은 인증기관용 DN의 구조를 정의하는 OpenSSL 설정화일의 예이다.

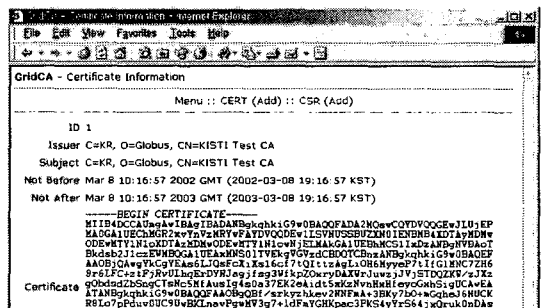
구현환경으로 PHP[6]가 설치된 웹서버를 이용하였다. 사용자가 입력한 정보와 비밀번호를 OpenSSL 명령에 전달하여 인증서에 관련된 처리가 이루어 지도록 하고, 그 결과 생성된 결과 화일을 읽고 파싱하여 데이터베이스에 저장하도록 구현하였다. 그림7은 인증기관(CA)의 인증서를 만들기 위해 정보를 입력하는 화면이고, 그림8은 결과 만들어진 CA의 인증서의 내용이다. 생성된 인증서와 비밀키는 데이터베이스에 저장된다. 사용자는 키와 인증요청서를 만들고, 이것을 CA가 사인해 줌으로써 사용자 인증서를 얻게 되는데, 그림9는 사용자가 자신의 정보를 입력하는 화면이고, 그림10은 그 결과 만들어진 인증요청서(Certificate Signing Request)의 내용이다.

```
[ req ]
default_bits            = 1024
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes

[ req_attributes ]

[ x509v3_extensions ]

# under ASN.1, the 0 bit would be encoded as 80
nsCertType                      = 0x40

[ req_distinguished_name ]
countryName             = Country Name
countryName_default     = KR
countryName_min         = 2
countryName_max         = 2

0.organizationName          = Organization Name
0.organizationName_default = Globus

commonName              = Your identify in your organization
commonName_max          = 64
```

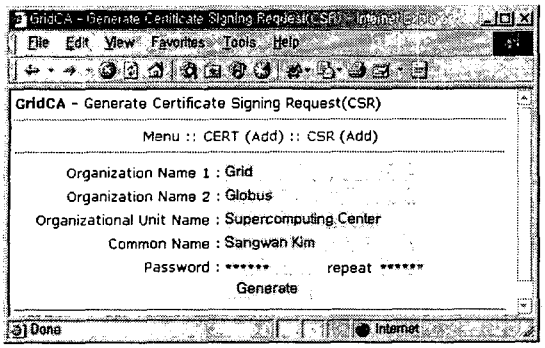
<그림6> 루트 인증기관용 DN구조 정의화일



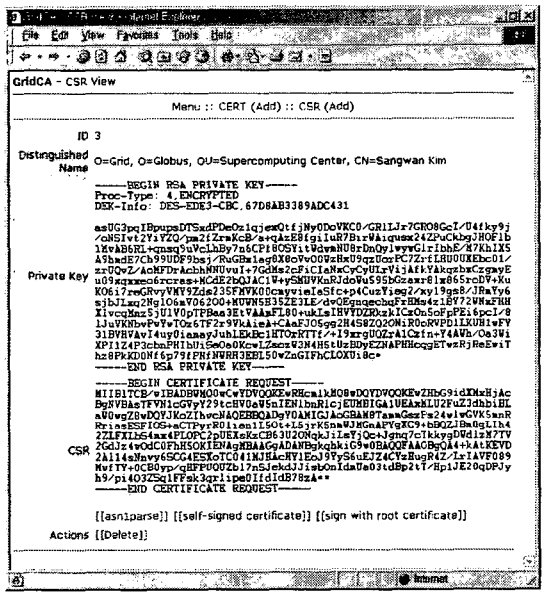
<그림7> CA 인증서 정보입력



<그림8> 생성된 CA 인증서



<그림9> 사용자 인증요청서 정보입력 화면



<그림10> 생성된 인증요청서(CSR)

인증요청서도 마찬가지로 데이터베이스에 저장되고, 관리자가 서명하는 과정을 거쳐 사용자 인증서가 만들어지게 된다. 게이트키퍼에 대한 인증요구서와 인증서도 이와 똑같은 과정으로 만들어진다.

4. 앞으로 연구 방향

현재 구현되어 있는 인증서 관리 도구는 매우 기본적인 기능만 구현되어 있으며, 관리자의 단독적인 사용만 가능하다. 향후에는 인증서 발급을 원하는 사용자는 누구나 자신의 인증서를 요청할 수 있도록 접근제어 기능을 추가할 예정이다. 또, CA가 서명하는 과정에서 관리자가 개입되는 것을 피하는 대신, 유효기간이 짧게 제한되어 있는 임시 인증서를 자동으로 발급할 수 있게 하는 기능도 추가할 예정이다.

그리고, 보다 근본적으로 OpenSSL의 컴맨드 라인 명령을 이용하지 않고, OpenSSL의 라이브러리를 직접 이용하여 구현할 것이다.

5. 결론

본 연구에서는 메타컴퓨팅 툴킷으로 많이 사용되고 있는 Globus를 직접 설치하고, 사용하려는 사람들에게 편리하고 체계적인 인증서 관리 방법을 제공해주는 웹기반 도구를 제작하였다.

현재 슈퍼컴퓨팅센터에 있는 슈퍼컴퓨터와 클러스터에 Globus가 설치되어 있는데[7], 모두 ANL의 인증기관에서 발행한 인증서를 사용하고 있고, 자체적으로 CA를 운영하고 있지 않기 때문에, 인증서와 키에 대한 관리가 제대로 이루어지고 있지 않는데, 본 연구와 향후연구에서 개발된 도구를 이용하여 이러한 문제를 해결할 수 있을 것으로 기대된다.

Globus는 현재 가장 많이 사용되고 있는 그리드 미들웨어이기는 하지만, 아직도 개선되어야 할 점이 많다. 특히 복잡한 설치과정과, 어려운 사용법 때문에 전산분야 전공자도 쉽게 설치하고, 시험운용하기 어려울 정도이다. 본 연구에서 개발된 도구를 이용하여 국내의 많은 연구기관에서 그리드와 Globus에 대한 연구가 활성화되기를 바란다.

참고문헌

[1] I. Foster and C. Kesselman, "The Grid: Blueprint for a New Computing Infrastructure", Morgan Kaufmann, 1998

[2] The Globus Project, http://www.globus.org

[3] I.Foster, C. Kesselman, G. Tsudik, S. Tuecke, "A Security Architecture for Computational Grids", Proc. 5th ACM Conference on Computer and Communications Security Conference, pg. 83-92, 1998

[4] OpenSSL(SSLeay), http://www.openssl.org

[5] Alfred J. Menezes, Paul C.van Oorschot, Scott A. Vanstone, "Handbook of Applied Cryptography", CRC Press, 1997

[6] PHP, http://www.php.com/docs.php

[7] 슈퍼컴퓨팅센터 2001년도 사업보고서, 2001, 한국과학기술정보연구원 슈퍼컴퓨팅센터