X.509 대리 인증서 환경에서 위임 추적 기능을 제공하는 ID 기반 암호 시스템 기반 권한 위임 프로토콜

(Privilege Delegation Protocol Providing Delegation Traceability Using ID-based Cryptosystem in X.509 Proxy Certificate Environment)

이 윤 호 † 김 병 호 ††
(Younho Lee) (Byungho Kim)

요 약 계산적 그리드 환경에서 개체간 권한 위임 및 Single Sign-on 의 목적을 위해 사용되고 있는 X.509 대리 인증서 표준은 추적 불가능성으로 인한 잠재적인 보안 위협 및 권한 위임자와 권한 대리자간의 다수의 대화식 (Interactive) 통신으로 야기 되는 비효율성에 노출되어 있다. 본 논문에서는 이러한 두 가지 문제점을 해결하면서 기존의 X.509 대리 인증서 표준의 장점을 그대로 유지할 수 있는 권한 위임 프로토콜을 제안한다. 제안 방법은 ID 기반 서명 알고리즘 및 키 생성 방법을 권한 위임 과정에 적용시켜 권한 대리키로 사용한다. 이러한 결과 권한 대리자와 권한 위임자간의 통신 횟수를 줄일 수 있다. 본 제안 프로토콜을 계산적 그리드 환경에 적용시키면, 연속된 위임 과정으로 생성되는 대리 인증서 사슬의 참여자를 알 수 있음으로써 생기는 보안성 향상 뿐만 아니라 광대역 네트워크상에서 진행되는 위임 과정의 통신량 및 횟수를 줄일 수 있으므로 결과적으로 계산적 그리드 환경의 성능 향상에 기여하게 된다.

키워드 : 그리드 보안, X.509 대리 인증서, 대리 서명, 정보 보호

Abstract Currently, the X.509 proxy certificate is widely used to delegate an entity's right to another entity in the computational grid environment. However it has two drawbacks: the potential security threat caused by intraceability of a delegation chain and the inefficiency caused by an interactive communication between the right grantor and the right grantee on the delegation protocol. To address these problems for computational grids, we propose a new delegation protocol without additional cost. We use an ID-based key generation technique to generate a proxy private key which is a means to exercise the delegated signing right. By applying the ID-based key generation technique, the proposed protocol has the delegation traceability and the non-interactive delegation property. Since the right delegation occurs massively in the computational grid environment, our protocol can contribute the security enhancement by providing the delegation traceability and the efficiency enhancement by reducing the inter-domain communication cost.

Key words : Grid security, X.509 Proxy Certificate, Secure Delegation, Proxy Signature

† 정 회 원 : 한국과학기술원 정보전자연구소 박사후연구원
yhlee@nslab.kaist.ac.kr
†† 정 회 원 : 경성대학교 컴퓨터공학과 교수
bkim@ksu.ac.kr
논문접수 : 2006년 9월 20일
심사완료 : 2008년 6월 2일

정보과학회논문지: 시스템 및 이론 제35권 제9호(2008.10)

1. 서 론

계산적 그리드(computational grid) 환경은 다양한 형태의 컴퓨터, 데이타 저장기기 및 그 외의 기기들을 광대역 네트워크로 연결하여 대규모 분산 계산을 가능하게 한다[1,2]. 계산적 그리드는 분산 슈퍼 컴퓨팅, 다양한 컴퓨터 기반 정밀 기기 및 분산 데이타 마이닝 (distributed data mining) 등의 진보된 응용 분야에 적용이 가능하다. 그리드 환경에서는 관리 영역이 상이한

위치의 자원, 예를 들어 연산 장치, 저장 장치, 네트워크 대역 등에 대한 병렬적, 분산적 사용을 필요로 한다. 이러한 각각의 관리 영역에서는 개별적인 보안 방법의 존재가 가능하다. 따라서 각 사용자는 자신이 원하는 자원의 사용을 위하여 해당 자원이 속한 영역의 관리자에게 인증 및 권한인가를 받아야만 자원을 사용할 수 있다. 또한 그리드 환경에서 자원의 사용 주체는 사용자가 아닌, 사용자의 작업을 수행하는 프로세스이다. 즉, 프로세스가 작업 수행 시 필요한 자원들을 사용하게 되므로, 사용자는 자신이 원하는 작업을 수행하는 프로세스에게 필요한 자원을 사용할 수 있는 권한을 위임해 주어야 한다. 그러나 그리드 환경은 기존의 서버-클라이언트(Server-Client) 환경과는 달리 프로세스가 사용해야 하는 자원은 다양한 관리 영역에 존재하므로 기존의 방법과는 다른 새로운 권한 위임 방법이 필요하게 되었다.

이러한 그리드 환경에서의 안전한 권한 위임 기능의 구현을 목적으로 X.509 대리 인증서(Proxy Certificate) 개념이 제안되었다[3,4]. 대리 인증서는 ITU-T X.509 공개키 기반 구조(공식명칭은 CCITT X.509 및 ISO/IEC/ITU 9594-8)의 확장으로써, X.509 공개키 인증서를 갖는 개체가 자신의 권한에 대한 부분적 또는 전체적인 위임을 가능하게 한다. 또한 대리 인증서를 이용하여 위임 개체 및 다른 개체간에 인증 및 보안 채널 생성이 요구될 경우, 일반 X.509 최종 개체 인증서와 같은 방법으로 인증 및 보안 채널 생성이 가능하다.

X.509 대리 인증서 표준에서의 권한 위임 방법은 전자 서명 및 인증서에 기반한 객체간 권한 위임 방법을 다룬 1993년의 Neuman의 제안 방법을 근거로 설계되었다[5]. 대리 인증서는 권한 대리자가 대리 서명에 쓰일 공개키/개인키 쌍을 생성한 후, 권한 위임자가 서명을 통하여 공개키에 대한 인증서를 생성, 전달하는 방식으로 이루어져 있다. 또한 권한 위임자가 권한 소유자일 경우, 대리 인증서는 권한 소유자의 X.509 개체 개인키의 서명으로서 발급되며, 그렇지 않은 경우, 권한 위임자는 자신이 권한을 위임받을 때 새로 생성한 개인키를 이용하여 서명함으로써 대리 인증서가 발급된다. 그림 1은 X.509 대리 인증서 표준의 권한 위임 방법에 대한 대략적인 설명을 나타낸다.

그러나, 본 방법은 권한 대리자가 기존의 권한 대리자의 X.509 개체 개인키와 독립적인 새로운 개인키/공개키 쌍을 생성해야 하기 때문에 다음과 같은 효율성/안전성 측면의 단점을 야기한다. 효율성 측면의 단점으로, 권한 위임 과정이 권한 위임자와 권한 대리자간의 다수번의 통신으로 이루어진다는 것이다. 현재의 X.509 대리 인증서 표준에서는 권한 대리자가 생성한 새로운 키를 권한 위임자에게 검증받기 위해 그림 1의 3번 과정을

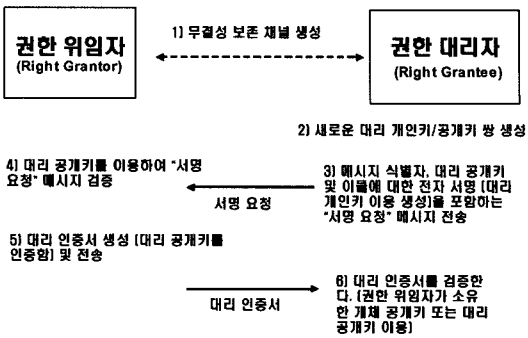

그림 1 X.509 대리 인증서 표준에서의 권한 위임 프로토콜

필요로 한다. 따라서 현재의 위임 방법은 다수의 양방향 통신을 필요로 하는 쌍방향 대화식 방법이며 위임 과정 시마다 이전 상태의 정보를 유지해야 되는 큰 비용을 갖게 된다. 안전성 측면의 문제로써, 현재의 대리 인증서 표준은 연속된 위임의 결과로써 발생되는 대리 인증서 사슬로부터 대리 인증서 사슬에 참여자 사용자의 신원을 확인할 수 없는 단점이 존재한다. 현재의 위임 프로토콜의 결과로써 발생되는 대리 인증서의 사슬에서 단지 권한 소유자만이 X.509 개체 공개키가 드러나고, 나머지 대리 인증서는 모두 권한 대리자가 새로 생성한 공개키만이 드러남으로 인하여, 권한 소유자의 신원만을 파악할 수 있고 대리 인증서 사슬의 나머지 개체에 대한 신원의 파악이 불가능하다. 이것은 X.509 대리 인증서 표준의 논평(commentary) 부분에서 언급한 바와 같이, 대리 인증서 사슬의 중간 단계의 참여자가 공격을 당하여 해당 참여자의 접근을 배제시켜야 하는 경우에도 해당 참여자의 신원을 확인할 수 없는 관계로 서비스 배제를 어렵게 만든다.

본 연구에서는 위임 사슬 참여자에 대한 신원 확인이 가능하고, 권한 수여자로부터 권한 대리자로의 1회만의 통신으로 위임을 구현할 수 있는 방법을 제안한다. 본 방법은 권한 확인자가 대리 인증서 사슬의 각 참여자의 신원을 확인할 수 있으므로 악의적인 참여자가 사슬의 중간 단계에 존재하는 권한(특정 서비스 또는 자원) 요청에 대하여 거부할 수 있으므로 대리 인증서 환경의 보안성을 향상시킨다. 이것은 권한 위임 과정에서 권한 대리자가 생성하는 키가 기존의 X.509 개체 개인키와 특정한 관계를 갖게 함으로써, 추후 대리 인증서 검증시 대리 인증서 발급자의 개체 공개키를 알아야만, 대리 인증서를 검증할 수 있는 대리 공개키를 알아낼 수 있도록 한 것에 기인한다. 이러한 이유로 제안 프로토콜에서는 대리 인증서 발급시 권한 위임자와 권한 대리자의 개체 공개키가 대리 인증서 내부에 기술되며 추후 위임 인증서 사슬에 기록된 공개키를 이용하여 위임 인증서

사슬에 참여한 사용자의 신원을 확인할 수 있다. 또한 본 방법은 인증 과정을 제외하고, 권한 대리자로부터 권한 위임자로의 통신이 필요없는 장점이 존재한다. 이것은 권한 위임 프로토콜을 비 대화식으로 만들며 이것은 권한 위임 프로토콜 수행 과정에서 권한 위임자 및 권한 대리자가 관리하는 상태 정보의 양을 감소시키므로 권한 위임 프로토콜의 효율성을 증대시킨다.

본 연구에서 제안한 새로운 방법은 암호학 분야에서 제안된 ID기반 서명 알고리즘을 이용하여 구현된다. ID 기반 서명 알고리즘은 사용자의 ID에 대응하는 개인키를 키 생성 센터로부터 안전한 채널로 전송받아 해당 개인키를 이용하여 서명을 수행하고, 서명 검증자는 서명자의 ID와 키 생성 센터의 공개키만을 알면, 서명자의 공개키를 생성할 수 있다. 따라서 서명 검증자는 따로 서명자의 공개키의 유효성을 검증할 필요가 없는 장점이 존재한다.

이전 연구와 구별되는 본 연구의 특징은 이러한 ID기반 서명 알고리즘의 사용자 키 생성 과정을 위임과정에 적용시킨 것이다. 권한 위임 수행시에, 권한 위임자는 권한 대리자에게 서명권한 기술 및 권한 대리자의 공개키를 포함하는 대리 인증서를 권한 대리자에게 전달한다. 그 후 권한 대리자는 해당 서명 권한 기술을 ID로 하는 ID기반 개인키를 생성하여서 위임 받은 권한에 대한 서명 작업을 수행한다. 이렇게 생성된 서명에 대하여 서명 검증자는 대리 인증서에 포함된 서명 권한 기술 및 권한 대리자의 공개키를 이용하여 권한 대리자의 ID 기반 개인키에 대응되는 ID 기반 공개키를 생성한 후, 권한 대리자의 서명에 대한 검증을 수행한다. 결과적으로 ID기반 서명 시스템에서 키 생성 센터가 권한 대리자로 대응되며, ID 기반 서명 시스템의 각 사용자의 개인키는 위임된 권한마다 생성되는 권한 대리자의 새로운 대리 개인키로 대응된다. 또한 ID는 권한 위임자가 권한 대리자에게 발급한 대리 인증서 정보로 대응된다.

본 논문의 제안 위임 알고리즘은 임의의 ID기반 서명 시스템으로 구현이 가능하다. 현재 알려진 효율적인 ID-기반 서명 알고리즘으로 Cha-Cheon 알고리즘 및 Hess 등의 알고리즘이 있다[6,7]. 본 논문에서는 Cha Cheon 서명 알고리즘[6]을 이용한 제안 위임 프로토콜과 기존의 RSA 알고리즘을 이용한 위임 프로토콜의 성능을 비교한다.

본 논문의 구성은 다음과 같다. 제2절에서는 X.509 대리 인증서 표준에 대해서 기술하고, 현 표준의 문제점에 대해 논의한다. 제3절에서는 제안 방법을 소개하고, 제4절에서는 현재의 X.509 대리 인증서 표준에서의 위임 프로토콜과 제안 프로토콜과의 성능 비교를 수행한다. 마지막으로 제5절에서 결론을 맺는다.

2. X.509 대리 인증서 표준

X.509 대리 인증서 표준은 계산 그리드 환경을 연구 및 구현하고 있는 세계적인 과제인 Globus Project [8]에서 제안된 GSI(Grid Security Infrastructure)에서 구현되었고, 이것을 표준화한 내용이다. GSI는 현재 다양한 미들웨어 라이브러리 및 어플리케이션에 사용되고 있다. 본 절의 구성은 다음과 같다. 제1 세부 절에서는 제정 동기를 기술하고, 제2 세부 절에서는 표준에서 제시하고 있는 권한 위임 프로토콜에 대해 기술한다. 제3 세부 절에서는 권한 위임 프로토콜의 특징에 대해 기술하며, 마지막 제4 세부 절에서 현 표준의 문제점을 지적한다.

2.1 제정 동기

X.509 대리 인증서 표준의 제정 동기는 다음의 예와 같은 환경에서의 필요성을 근거로 한다. 본 예는 X.509 대리 인증서 표준에 기술되어 있는 부분이다[3]. 그림 2는 아래의 내용에 대한 설명을 묘사한다.

"Steve는 그의 회사의 Intranet 기반 Grid 위의 다양한 호스트들 사이에 많은 수의 대용량 파일의 이동을 관리하는 안정적인 파일 이동 서비스를 사용하고자 하는 기술자이다. 그의 노트북에서, 그는 다른 off-line 작업들을 수행하면서 대용량의 파일들을 이동시키고자 한다. 파일 전송 서비스는 전송을 시작하기 전에 파일 이동 요청을 큐에 저장에 놓을 수 있다. 이후에 파일 전송 서비스는 원본 파일이 저장되어 있는 호스트와 전송 대상 호스트에 각각 연결하여 원 호스트와 대상 호스트사이에 직접적으로 파일 송/수신이 수행되도록 한다. 스티브는 전송 중간에 노트북에서 Agent를 수행하여 파일 전송 상태를 확인할 수 있다. 그는 모든 이러한 과정이 그의 회사의 자원을 보호하면서 안전하게 수행되기를 원한다. 이때에 회사의 자원은 Steve의 PKI 기반의 그의 스마트카드에 있는 자신의 X.509 개인키를 통하여서만 초기화될 수 있다."

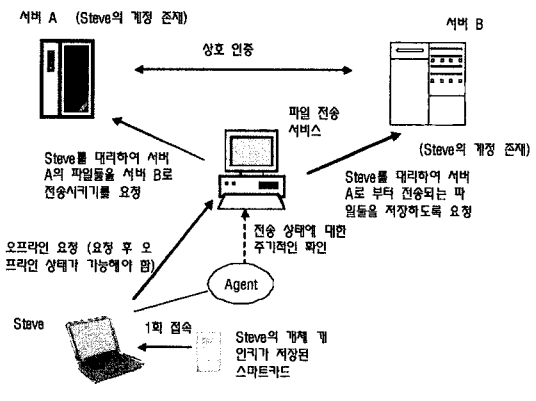

그림 2 X.509 대리 인증서 사용 예로써의 Grid 환경

위와 같은 시나리오 중에, 다음과 같은 곳에서 인증 및 권한 위임을 필요로 한다.

1) Steve는 그의 파일 전송 요청을 파일 전송 서비스에 제출하기 위해 파일 전송 서비스와 Steve간의 상호 인증이 필요하다.
2) 파일 저장 호스트들은 파일 전송 서비스에 대하여 어떠한 정보도 갖고 있지 않기 때문에, 파일 전송 서비스는 파일 소유자 또는 파일 저장 호스트의 사용자로부터 권한을 위임 파일 저장 호스트와의 상호 인증 및 파일 전송을 수행할 수 있는 권한을 얻어야 한다.
3) 원 파일을 소유하고 있는 호스트와 파일 전송 대상 호스트는 파일이 올바른 곳으로부터 전송되고, 또한 파일전송 서비스가 지정한 대상 호스트로 파일 전송이 이루어지는 것을 확인하기 위해 상호 인증이 필요하다.
4) Steve의 노트북에서 수행되는 Agent는 전송 결과를 확인하기 위해 파일 전송 서비스와 상호 인증을 필요로 한다.

X.509 대리 인증서는 위의 4가지 경우에 대하여 야기되는 두 가지 관련 문제를 해결할 수 있다.

1) Single sign on: Steve는 그의 스마트카드를 노트북에 1회만 연결하여 패스워드 1회 입력만으로 모든 파일 전송 요청을 파일 전송 서비스에 제출하기를 원한다. 또한 이후에 주기적으로 전송 상태 확인 시에는 스마트카드 연결 없이 작업 수행이 가능하기를 원한다.
2) Delegation: 위의 시나리오에서는 Steve를 대리하여 다양한 원격지 프로세스가 동작하게 된다. 따라서 해당 프로세스들은 Steve를 대리하기 위한 위임 권한을 필요로 한다. 예를 들어, 파일 전송 서비스는 Steve를 대리하여 각 파일 호스트들과 상호 인증을 수행해야 한다. 이 때에 파일 전송 서비스는 Steve로부터 권한을 위임받아서 상호 인증 과정을 대리하여 수행할 수 있다.

2.2 위임 프로토콜

X.509 대리 인증서(PC: Proxy Certificate)는 X.509 PKI 기반 인증 환경에서 제한된 권한 대리 기능을 제공한다. 대리 인증서는 다음과 같은 성질을 갖는다.

1) 대리 인증서는 X.509 개체 인증서(EEC: End-Entity Certificate) 또는 다른 대리 인증서에 의해 서명된다. 이러한 서명하는 주체를 발급자(PI: Proxy Issuer)라 한다.
2) 대리 인증서는 단지 또 다른 대리 인증서에 대한 서명을 위해 사용이 가능하고 개체 인증서에 대한 서명은 불가능하다.
3) 대리 인증서는 자신만의 개인키/공개키 쌍을 갖고 이것은 다른 대리 인증서와 개체 인증서의 것과는 구별된다.
4) 하나의 대리 인증서가 발급한 대리 인증서들 또는 동일한 개체 인증서를 인증 사슬의 최상위로 갖는 대리 인증서들의 식별자(ID)는 각각 유일하며, 각 대리 인증서 안에는 발급자가 제공하는 (제한된) 권한에 대한 기술이 존재한다.
5) 대리 인증서는 자신이 대리 인증서임을 확인시킬 수 있는 X.509 확장을 포함한다.

위의 특징을 갖는 대리 인증서는 다음과 같은 과정으로 생성된다.

1) 새로운 개인키/공개키 쌍을 생성한다.
2) 1)에서 생성된 키 쌍은 대리 인증서 요청 생성에 사용된다.
3) 개체 인증서나 다른 대리 인증서의 개인키를 이용하여 서명된 새로운 대리 인증서가 요청에 대응하여 생성된다. 이러한 과정 동안, 대리 인증서 생성 요청이 발급자에 의하여 검증된다.

생성된 대리 인증서는 아래의 그림 3과 같은 형식을 갖게 된다. 대리 인증서에 기술되는 대리 공개키 정보는 기본 인증서 영역에 기재되나, 그 외의 대리 인증서에 특화된 정보는 확장 인증서 영역에 기재된다. 발급자의 서명은 대리 인증서의 전체 영역을 메시지 부분으로 하여 생성된다. 대리 인증서는 개체 인증서와는 다른 확장 인증서 영역을 갖기 때문에, 인증서의 형식 만으로 두 인증서에 대한 구분이 가능하다.

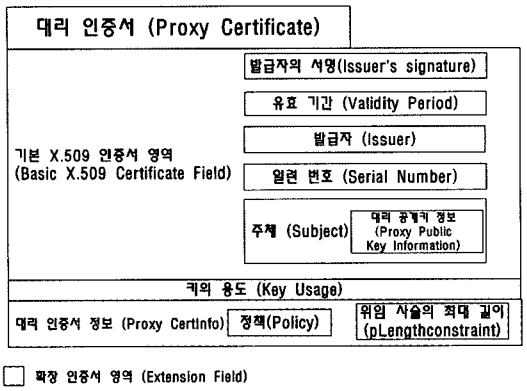

그림 3 대리 인증서의 형식

2.3 프로토콜의 특징

위와 같은 위임 프로토콜로써 생성되는 대리 인증서 방식은 다음과 같은 기능을 갖는다.

• 역동성(Dynamic) : 사용자의 권한을 위임 받아 작업

을 수행하는 프로세스는 작업 수행 중에 다양하게 생성/제거 될 수 있다. 그리고 프로세스가 사용하는 자원도 수행 도중에 자원 소유자의 의도에 의해 생성/제거 될 수 있다.

- 경량성(Light-weight) : 하나의 작업에도 많은 권한 위임이 발생하고, 다양한 사용자들이 존재하기 때문에 권한 위임 과정은 계산량 및 네트워크 사용량 측면에서 적은 자원을 사용해야 한다.
- 제한된 위임(Restricted delegation) : 사용자의 권한을 위임 받는 프로세스는 자신이 수행할 작업에 관련된 자원만 사용이 가능하게 하는 기능을 갖는다. 그 이상의 권한을 위임 받을 경우, 차후 프로세스가 공격자에게 노출당했을 경우, 더 많은 피해를 입을 수 있다.
- 반복적 위임(Repeated delegation) : 사용자의 작업을 수행할 주 프로세스는 자신의 작업을 분배하여 다수의 부 프로세스에게 작업을 나눠줄 수 있다. 이 경우, 주 프로세스는 사용자로부터 위임받은 자신의 권한을 부 프로세스에게 재위임할 수 있다. 이러한 재위임은 반복적으로 수행이 가능하다.

또한 권한 대리자의 X.509 개체 개인키에 위임하지 않고 새로운 키를 생성함으로써 다음과 같은 보안 관련 특성을 갖는다.

- 개체 개인키의 보호(Protection of entity private key): 권한 대리자의 X.509 개체 개인키에 과도한 권한이 집중되는 것을 방지하여, 공격자가 권한 위임키를 획득하게 되더라도 그에 대한 피해를 권한 위임자가 위임한 권한으로 제한할 수 있다.
- 파기의 용이성(Ease of revocation): 추후에 위임된 권한이 파기(Revocation)되었을 때, 그에 대한 처리 과정이 단순화 되는 장점이 있다. 만약 권한 대리자의 X.509 개체 개인키에 위임 권한을 부여하여 그것이 파기될 경우, X.509 개체 개인키는 여전히 유효한 키이고 파기된 내용은 위임 권한에 한하기 때문에, 해당 위임 권한에 대한 파기 리스트를 권한 대리자의 개인키 자체가 파기될 때까지 관리를 해주어야 한다. X.509 개체 개인키는 유효기간이 길기 때문에, 이러한 파기 리스트 관리 비용은 위임 권한이 많아질수록 증가하며 그 관리는 개인키의 유효기간까지 지속되어야 하므로 그 비용이 크다. 이에 반하여 상대적으로 유효기간이 짧은 새로운 키를 생성하여 그에 대하여 위임 권한을 부여한 경우, 위임 권한이 종료될 시에 새로 생성된 키를 동시에 파기하면 되므로, 상대적으로 파기 리스트 유지비용이 적은 장점이 존재한다.

2.4 현 X.509 대리 인증서 표준의 권한 위임 방식의 문제점

현재의 제안 표준은 다음과 같은 두 가지 문제점을

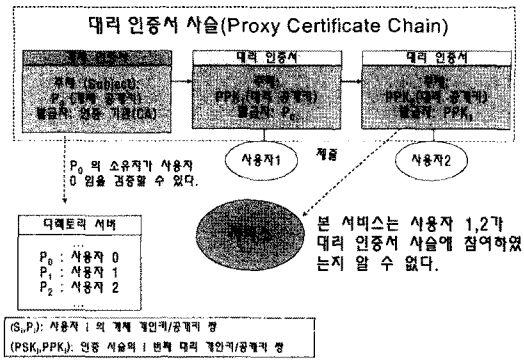

그림 4 위임 추적 불가능성 문제

지니고 있다. 하나는 대리 인증서 사슬의 참여자가 누구인지 알 수 없기 때문에 발생하는 위임 추적 불가능성(Delegation Intraceability)이고, 두 번째 문제는 현재의 권한 위임 프로토콜이 On-line 프로토콜임으로써 야기되는 비효율성이다.

첫 번째 문제점은 대리 인증서의 사슬의 추적 불가능성이다. 현재의 표준에서는 대리 인증서 생성시, 권한을 위임 받는 권한 대리자는 위임 받는 권한 행사를 위해 자신의 인증을 위해 사용되는 개체 대리키/공개키와 아무 관계가 없는 새로운 대리 개인키/공개키를 생성하고, 이를 대리 인증서에 기재한다. 그리고 권한 대리자는 자신이 위임받은 권한의 사용을 위해 특정 서비스에 접근할 경우, 해당 서비스에 단지 자신이 소유한 대리 인증서의 사슬 만을 제출한다. 따라서 서비스의 입장에서는 권한 대리자의 신원 확인이 불가능 하며, 단지 대리 인증서를 제출한 개체가 권한 소유자로부터 적법하게 권한을 위임 받았는지에 대한 확인 만이 가능한 것이다. 만약 권한 대리자가 자신의 개체 인증서를 서비스에게 동시에 제출하여도 상황은 변하지 않는다. 왜냐하면, 대리 인증서 내부에, 해당 대리 인증서를 소유하게 되는 권한 대리자의 신원을 검증할 수 있는 정보 (예를 들어, 권한 대리자의 개체 개인키로 서명한 서명 정보 등)이 전혀 포함되지 않기 때문이다.

그러나 특정 서비스가 대리 인증서를 유효성을 검증하고자 할 때, 본 대리 인증서를 포함하는 인증서 사슬에서 누가 인증서 발급에 참여하였는지를 아는 것은 중요하다. 예를 들어, 특정 서비스 또는 자원이 도용되어 해당 서비스 또는 자원의 접근을 배제하여야 할 경우가 존재 할 수 있다. 이러한 경우, 만약 대리 인증서 사슬 중의 일부분이 도용된 서비스 또는 자원인지 확인 할 수 있다면, 해당 대리 인증서를 배제시킬 수 있을 것이다. 그러나 현재의 대리 인증서 사슬에는 원래의 권한 소유자의 개체 인증서 공개키만이 드러나고, 나머지는 모두

위임 과정시 생성한 새로운 공개키만이 드러나게 된다. 따라서 대리 인증서 사슬의 최상위 발급자의 신원만 확인되며 나머지 참여자의 신원은 확인할 수 없는 단점이 있다. 위의 그림 3은 본 문제를 요약한 내용이다.

두 번째 문제점은 현재의 프로토콜이 온라인(On-line) 프로토콜이 아니라는 것이다. 현재의 권한 프로토콜은 권한 위임자와 권한 대리자간의 인증 및 권한 위임 과정이 분리되어 있다. 이것은 권한 위임자가 권한 대리자의 새로 생성한 키를 검증하고, 그에 대한 대리 인증서를 발급하는 과정 때문에 발생한다. 이러한 인증 및 권한 위임 과정의 분리는 전체 위임 프로토콜을 오프라인(Off-line) 프로토콜이 아닌 온라인 프로토콜로 만들며 이러한 이유로 위임 프로토콜 수행 시에 session 관리 비용 및 통신 비용 등이 증가한다. 특히 계산적 그리드 환경에서는 이러한 위임 프로토콜이 광대역 네트워크를 통하여 수행되기 때문에 온라인 프로토콜은 전체 위임 프로토콜의 성능에 치명적인 영향을 미치게 된다.

본 논문은 위와 같은 현재의 X.509 대리 인증서 표준의 문제점을 해결하면서, 동시에 현재 표준이 갖고 있는 장점은 유지시키는 새로운 대리 인증서 방식의 위임 프로토콜을 제안하는 것을 목표로 한다.

3. 제안 방법

본 절에서는 이전 절에서 제시한 문제점을 해결한 새로운 방법을 제안한다. 본 절의 구성은 다음과 같다. 제 1 세부 절에서는 ID기반 암호 시스템의 개념에 대해 설명하고, 제2 세부 절에서는 제안 방법을 기술한다.

3.1 ID 기반 암호시스템

ID 기반 암호시스템 개념은 1984년 Shamir에 의해 제안되었다[9]. ID 기반 암호 시스템은 전송 대상자의 이메일 주소, IP 주소와 같은 ID 정보만으로 사용자의 공개키를 알 수 있음으로써, 현재의 PKI 기반 암호시스템의 문제점인 공개키 유효성 검증 문제를 쉽게 해결할 수 있는 장점이 있다. ID 기반 암호시스템은 키 생성 센터와 일반 사용자가 존재한다. 키 생성 센터는 자체의 개인키/공개키 쌍을 갖고 있고, 이 중 공개키 값은 널리 알려진 값이라고 가정한다. 일반 사용자는 자신만의 고유한 ID를 갖고 있으며, 키 생성 센터(KGC)로부터 자신의 ID에 해당되는 개인키를 보안 채널을 통해 발급받는다. 본 논문에서 관심을 두고 있는 것은 ID기반 암호시스템을 이용한 서명 방법이다. 본 ID기반 암호시스템의 서명 알고리즘에서 필요한 알고리즘은 다음의 4가지이다.

1) ID기반 개인키 생성 알고리즘 (IDKeyPrv): ID 및 키 생성 센터의 개인키를 입력값으로 하여, 해당 ID에 대응되는 ID기반 개인키를 생성한다.
2) ID기반 공개키 생성 알고리즘 (IDKeyPub): ID 및 키 생성 센터의 공개키를 입력값으로 하여, 해당 ID에 대응되는 ID기반 공개키를 생성한다.
3) ID기반 서명 알고리즘 (IDSign): 서명 대상 메시지와 ID기반 개인키를 입력값으로 하여 서명값을 생성하는 알고리즘이다.
4) ID기반 서명 검증알고리즘 (IDVrfy): 서명 대상 메시지와, 서명값, 그리고 서명자의 ID에 대응되는 ID 기반 공개키를 입력값으로 하여, 해당 서명이 ID에 대응되는 사용자의 개인키로 생성한 것과 입력된 서명 대상 메시지를 서명한 것인지를 검증한다.

그림 5는 ID 기반 서명 시스템의 대략적인 수행 과정을 나타낸다. 우선 ID_0를 갖고 있는 User 0가 자신의 아이디를 키 생성 센터(KGC)에게 전달하면, 키 생성 센터는 ID_0에 해당 되는 ID기반 개인키 $s(ID_0)$를 생성하고, 그것을 안전한 채널을 통해 User 0에게 전달한다. 이후 User 0가 ID기반 서명을 수행하고 싶다면 자신이 갖고 있는 ID기반 개인키를 이용하여 서명한다. 서명 검증자는 서명 정보 및 서명자의 ID 정보를 이용하여 서명자가 갖고 있는 ID 기반 공개키를 생성하고, 그것을 이용하여 서명을 검증한다.

이러한 ID 기반 서명 알고리즘을 구현한 대표적인 예로는 Cha-Cheon 방법[6], Paterson의 방법[10] Hess의 방법[7] 등이 존재한다.

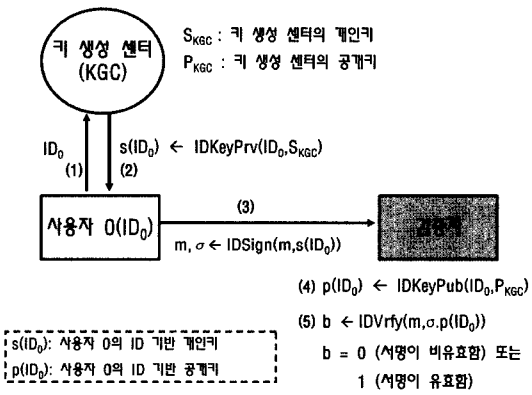

그림 5 ID 기반 전자 서명 시스템의 개괄

3.2 제안 프로토콜

본 세부 절에서는 제안 프로토콜에 대해 기술한다. 제안 프로토콜은 기존의 표준과는 달리, 일반 서명 알고리즘과 ID 기반 서명 알고리즘을 동시에 이용한다. 제안 프로토콜은 기존의 권한 대리자의 개인키/공개키 쌍과 독립적인 새로운 키를 생성하는 대신, 권한 위임자가 권한 대리자에게 주는 대리 인증서 정보와 권한 대리자가

생성하는 난수를 이용하여, 권한 대리자의 개인키의 정보를 포함되는 새로운 개인키/공개키를 생성하는 것이다. 이러한 방식으로 생성된 개인키/공개키쌍은 권한 대리자의 개체 개인키/공개키와 특정한 관계를 갖고, 특히 새로 생성된 공개키는 대리 인증서 정보와 권한 대리자가 사용한 난수 및 권한 대리자의 개체 인증서에서 제공하는 개체 공개키를 이용하여 만들 수 있다. 제안 프로토콜은 기존 방법과는 달리 권한 대리자가 생성한 새로운 공개키 정보 및 이에 대한 전자 서명을 권한 위임자에게 보낼 필요가 없기 때문에 전체 통신 과정이 대리 인증서 전달을 위한 권한 위임자로부터 권한 대리자로의 1회의 일방향 통신만으로 위임이 끝나게 된다. 또한 위임 권한의 사용을 위하여 새로 생성된 개인키를 이용하여 서명을 생성할 경우, 권한 검증자는 권한 대리자의 개체 공개키를 이용하여 새로 생성된 개인키에 대응하는 공개키를 생성한다. 따라서 대리 인증서 사슬에는 위임 과정에 참여한 개체들의 개체 공개키 정보가 포함되며, 이를 이용하여 권한 검증자는 대리 인증서 사슬에 참여한 개체들의 신원을 확인할 수 있다. 또한 대리 권한 사용을 위해 생성된 개인키로부터 권한 대리자의 개체 개인키를 알아내는 것은 매우 많은 계산량을 필요로 하기 때문에 알아내기 어렵다. 따라서 기존 표준의 방법과 마찬가지로 위임된 권한을 사용할 때에 권한 대리자의 개체 개인키는 보호된다.

3.2.1 사용되는 기호들

본 제안 프로토콜의 기술을 위해 사용되는 기호는 다음 표 1과 같다.

3.2.2 제안 프로토콜

제안 권한 위임 프로토콜은 다음과 같이 기술된다. 먼저 제안 프로토콜의 환경은 다음과 같다. 권한 위임 프로토콜은 권한 위임자가 O인 경우와 A인 경우의 두 가지로 나뉘며, 개체 B가 권한을 수여 받는다고 가정한다. 제안 프로토콜은 권한 위임의 경우와 권한 행사의 경우로 나뉘어진다. 권한 위임자가 O인 경우는 자신의

표 1 사용되는 기호들

기호	설명
O, A, B	각각 권한 소유자, 다른 개체로부터 권한을 위임받은 개체, 권한을 위임받아서 권한을 행사하는 개체를 나타낸다.
s_O, s_A, s_B	권한 소유자 및 개체 A, B 각각의 개체 개인키.
p_O, p_A, p_B	권한 소유자 및 개체 A, B 각각의 개체 공개키 이다 디렉토리 서버 등을 이용하여 공개키 소유자에 대한 신원 확인이 가능하다.
$s_X(M)$	M을 ID로 하고, X의 개인키를 키 생성에 이용한 ID기반 개인키
$p_X(M)$	M을 ID로 하고, X의 개인키를 키 생성에 이용한 ID기반 공개키
$Sign(m,s)$	일반 전자 서명 알고리즘을 나타낸다. 입력 메세지 m에 대하여 개인키 s를 이용하여 서명을 수행한다. 결과로 전자 서명 값 σ가 나온다.
psk_X/ppk_X	대리 인증서로부터 검증받아 위임 받은 권한을 행사할 때 사용하는 X의 대리 개인키/공개키 쌍
$Vrfy(\sigma,m,p)$	전자 서명 검증 알고리즘을 나타낸다. 만약 σ가 메세지 m에 대하여 p에 대응하는 개인키를 이용하여 서명했는지 확인한다. 만약 서명이 올바르다면 1을, 그렇지 않은 경우 0을 결과값으로 돌려준다.
$m_{w(X)}$	권한 대리자의 공개키 정보 및 서명 정보를 제외한 대리 인증서의 모든 정보를 나타낸다. 이 값에는 위임되는 권한 내용도 포함되어 있다.
$IDSign(m,s(ID))$	ID 기반 서명 알고리즘을 나타낸다. 입력값으로 서명 대상 메세지 m 및 ID기반 개인키 $s(ID)$가 들어가고 결과로써 ID기반 서명값 λ가 생성된다.
$IDVrfy(m,\sigma,p(ID))$	ID 기반 서명 검증 알고리즘을 나타낸다. 입력값으로 서명 대상 메세지 m, ID 기반 서명 λ, 그리고 ID에 대응하는 ID 기반 공개키 $p(ID)$가 들어간다. 만약 서명이 ID 기반 공개키에 대응하는 개인키로 메시지 m에 대해 서명한 값일 경우 1을 반환하고 그렇지 않은 경우 0을 반환한다.
$IDKeyPrv(ID,s)$	ID 기반 개인키를 생성하는 알고리즘이다. ID 및 키 생성 센터의 개인키 s를 입력값으로 받아 ID에 대응하는 ID 기반 개인키 $s(ID)$를 반환한다.
$IDKeyPub(ID,p)$	ID 기반 공개키를 생성하는 알고리즘이다. ID 및 키 생성 센터의 공개키 p값을 입력값으로 하고, 결과로써 ID에 대응하는 ID 기반 공개키 $p(ID)$를 반환한다.
σ	$Sign$으로 생성되는 전자 서명값을 나타낸다.
λ	$IDSign$으로 생성되는 ID 기반 전자 서명을 나타낸다.
$A \leftarrow B$	A에 B값을 설정하는 것을 의미한다. B는 특정 함수가 될 수 있다. 함수일 경우 함수의 결과값이 A에 설정된다.
$>$, $=$, $<$, $\leq$, $\geq$	해당 기호 좌우에 기술되는 값의 비교를 나타내는데 쓰이는 기호이다.

개체 인증서를 권한 위임 사슬의 최상단에 위치시킨다.

권한 위임

A. 권한위임자가 O인 경우

1) O는 다음과 같은 대리 인증서 $Proxy Cert_{O\rightarrow B}$를 만들어 B에게 전송한다.

$Proxy Cert_{O\rightarrow B} \leftarrow (m_{w(B)}, p_O, p_B \sigma_{O\rightarrow B} = Sign(m_{w(B)} \| p_O \| p_B s_O))$.

2) B는 아래와 같은 과정을 수행하고 최종적으로 대리 권한 집행에 사용될 대리 개인 키 psk_B를 얻는다.

a) 대리 인증서 검증

(1) $b \leftarrow Vrfy(\sigma_{O\rightarrow B}, m_{w(B)} \| p_B, p_O)$.

(2) $b=0$인 경우 인증서를 폐기하고 위임 과정을 마친다.

b) 대리 개인키 생성

(1) 난수 n_B를 생성한다.

(2) $psk_B (= s_B(m_{w(B)} \| p_O \| p_B \| n_B))$

$(\leftarrow IDKeyPrv(m_{w(B)} \| p_O \| p_B \| n_B, s_B))$.

B. 권한 위임자가 A인 경우

1) A는 다음과 같은 대리 인증서 $Proxy Cert_{A\rightarrow B}$를 생성을 위해 아래와 같은 과정을 수행한다. A는 O로부터 $Proxy Cert_{O\rightarrow A} = (m_{w(A)}, p_O, p_A, \sigma_{O\rightarrow A} = Sign(m_{w(A)} \| p_O \| p_A, s_O))$를 받았고 이를 이용하여 대리 개인키 psk_A를 이미 생성했다고 가정한다. $Proxy Cert_{A\rightarrow B}$를 생성한 후, A는 B에게 $Proxy Cert_{O\rightarrow A}$, $Proxy Cert_{A\rightarrow B}$ 를 모두 전송한다.

(1) $\lambda_{A\rightarrow B} \leftarrow IDSign(m_{w(B)} \| p_A \| p_B, psk_A)$.

(2) $Proxy Cert_{A\rightarrow B} \leftarrow (m_{w(B)}, p_A, p_B, \lambda_{A\rightarrow B}, n_A)$.

2) B는 아래와 같은 과정을 수행하고 최종적으로 대리 권한 집행에 사용할 대리 개인키 psk_B를 얻는다.

a) 대리 인증서 검증

(0) $Proxy Cert_{O\rightarrow A}$의 검증을 위해 프로토콜 A-(a)의 과정을 수행한다.

(1) $(m_{w(B)}, p_A, p_B, \lambda_{A\rightarrow B}, n_A) \leftarrow Proxy Cert_{A\rightarrow B}$.

(2) $p_A(m_{w(A)} \| p_O \| p_A \| n_A) \leftarrow IDKeyPub(m_{w(A)} \| p_O \| p_A \| n_A, p_A)$.

(3) $b \leftarrow IDVrfy(m, \lambda_{A\rightarrow B}, p_A(m_{w(A)} \| p_O \| p_A \| n_A))$.

(4) $b=0$인 경우, 대리 인증서를 폐기하고 위임 과정을 끝낸다.

b) 대리 개인키 생성

(1) 난수 n_B를 생성한다.

(2) $psk_B (= s_B(m_{w(B)} \| p_A \| p_B \| n_B))$

$\leftarrow IDKeyPrv(m_{w(B)} \| p_A \| p_B \| n_B, s_B)$.

만약 B가 다른 개체에게 자신이 위임 받은 권한을 다른 개체 C에게 다시 위임 하고 싶다면, A에서 B로의 권한 위임에 사용된 프로토콜을 그대로 사용하면 된다. 단, 현재의 경우 $O \rightarrow A \rightarrow B$의 순서로 권한이 위임되어 왔으므로, B가 C에게 권한을 위임할 경우에는 권한 위임 프로토콜 B 수행시 생성되는 대리 인증서 $Proxy Cert_{B\rightarrow C}$ 뿐만 아니라, A로부터 전달 받았던 $Proxy Cert_{O\rightarrow A}$, $Proxy Cert_{A\rightarrow B}$도 모두 C에게 전해 주어야 한다. 또한 C는 대리 인증서 검증을 위한 프로토콜 (B-2) 부분) 수행시, 자신이 전달 받은 3개의 대리 인증서($Proxy Cert_{O\rightarrow A}$, $Proxy Cert_{B\rightarrow C}$, $Proxy Cert_{A\rightarrow B}$) 모두에 대해 검증을 수행하여야 한다. 이 경우 C는 B-2)-a)-(1)~B-2)-a)-(4)의 과정을 두 번 반복하게 된다.

위임 행사

만약 B가 위임 받은 권한을 사용하기 위해, 특정 메시지 m에 대하여 서명을 수행할 경우, B는 자신이 생성한 ID 기반 개인키를 이용하여 ID 기반 서명 λ_B $(= IDSign(m, psk_B))$을 생성한다. 그 후 서명 대상 메시지, 서명, 그리고 B가 자신의 대리 개인키를 생성하기 위해 생성한 난수인 (m, σ_B, n_B)와 B가 현재 갖고 있는 대리 인증서(또는 대리 인증서 사슬)를 모두 권한을 확인하는 권한 검증자(서비스 제공자 또는 자원 관리 호스트 등)에게 전달한다. 권한 검증자는 대리 인증서들을 모두 검증한 후, B가 생성한 서명을 아래의 과정을 통하여 검증한다. 본 과정은 B가 O로부터 권한을 위임받음을 가정한다.

1) $ppk_B (= p_B(m_{w(B)} \| p_O \| p_B \| n_B))$

$\leftarrow IDKeyPub(m_{w(B)} \| p_A \| p_B \| n_B, p_B)$.

2) $b \leftarrow IDVrfy(m, \lambda_B, ppk_B)$.

3) $b=1$인 경우, 전달받은 서명은 유효한 것이면 그렇지 않을 경우, 서명은 유효하지 않은 것이다.

3.2.3 제안 방법의 안전성

제안 방법과 표준에서 사용하는 일반 전자 서명 및 새로운 키 생성을 이용한 방법과의 차이점은 두가지이다. 하나는 서명 및 검증 과정에서 일반 전자 서명 생성 및 검증 알고리즘 뿐만 아니라 ID-기반 서명 생성 및 검증 알고리즘을 사용한 것이며 또한 기존 방법에서는 새로운 개인키/공개키 쌍을 생성하여 위임 과정시 해당 키를 대리 인증서에 연결하는 것과 달리 각 위임 과정마다 위임 정보를 ID로 하는 ID기반 개인키를 생성하여 사용한다는 것이다.

첫 번째 차이점에 대해 논의하면, 일반적인 전자 서명 알고리즘과 같이 ID기반 서명 알고리즘도 일반 전자 서명 알고리즘과 같이 서명 위조 불가능성 및 부인 방지성을 갖추어야 하므로 일반 서명 알고리즘 대신 ID 기반 서명 알고리즘을 사용함으로써 생기는 안전성의 차이점은 없다.

두 번째 차이점인 키 생성 관련 문제에 대해 논의하면, ID기반 암호 시스템에서는 각 ID에 대응하는 ID 기반 개인키로부터 키 생성 센터의 개인키를 알아낼 수 없는 성질을 반드시 갖추어야 한다. 따라서 원래의 표준과 같이 대리 권한의 사용을 위해 새로 생성한 키로부터 권한 대리자의 개체 개인키를 알아 낼 수 없어야 하는 성질을 만족한다. 또한 대리 권한과 대리 권한을 행사할 수 있는 대리 개인키와의 연결은 ID기반 암호시스템의 특성상, ID 및 키 생성 센터의 공개키를 알면 해당 ID에 대응하는 공개키를 알아 낼 수 있으므로, 권한 위임자는 권한 대리자가 ID 기반 키 생성 방법을 이용하여대리 개인키를 생성할 시에 ID 부분만을 인증해주면 된다. 제안 방법에서 ID 부분은 인증서 내용과 권한 대리자/권한 수여자의 공개키 정보이므로 본 정보에 대한 서명만을 대리 인증서에 추가한다면, 권한 대리자가 대리 개인키/공개키도 대리 인증서와 연결해 주는 것과 같은 효과를 얻게 된다. 현재의 ID기반 전자 서명 알고리즘(예, [6,7])는 본 조건을 만족한다.

결론적으로 위의 두 가지 논의로부터, 사용되는 ID기반 서명 알고리즘이 일반 서명 알고리즘과 비교하여 동등한 안전성을 갖는다면, 제안 위임 프로토콜과 표준의 위임 프로토콜 사이의 안전성의 차이는 존재하지 않는다는 것을 알 수 있다.

4. 성능 평가

본 절에서는 제안 프로토콜의 성능을 기존의 X.509 대리 인증서 프로토콜과 비교한다. 첫 번째 세부절에서는 제안 프로토콜과 표준 프로토콜의 기능성에 대한 비교를 수행하며, 두 번째 세부절에서는 현재 Globus Toolkit에서 사용되고 있는[11] RSA 알고리즘 기반 표준 권한 위임 프로토콜과 제안 방법을 이용한 권한 위임 프로토콜의 권한 위임을 위해 소요되는 시간을 비교한다.

4.1 기능성 비교

본 세부절에서는 제안 방법과 표준 프로토콜의 기능성에 대한 비교를 수행한다. 기능성 비교는 우리가 제시한 기능인 비대화식 위임(Non-interactive delegation) 및 위임 추적성(Delegation traceability)과 2.3 절에서 제시한 X.509 대리 인증서 표준의 6가지 특성인 역동성(Dynamic), 경량성(Lightweight), 제한된 위임(Restricted delegation), 반복 위임(Repeated delegation), 개체 개인키 보호(Protection of entity private key), 폐기의 용이(Ease of revocation)에 대하여 제안 프로토콜 및 현 표준 프로토콜이 이 해당 특성을 갖고 있는지의 여부에 대해 분석한다.

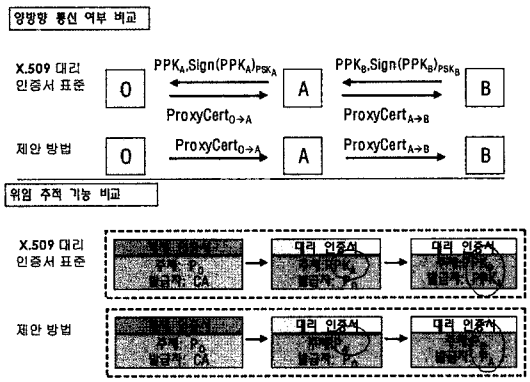

그림 6 제안 프로토콜과 X.509 대리 인증서 표준과의 비교

- 비대화식 위임: 그림 6의 윗부분은 권한 소유자 O가 A에게 권한을 위임하고, 또한 A는 B에게 권한을 위임하는 과정시에 O와 A, A와 B사이에 전달되는 정보를 나타낸 그림이다. 기존의 X.509 대리 인증서 표준의 위임 프로토콜은 권한 대리자가 권한 위임자에게 새로 생성된 키에 관한 정보를 보내고, 또한 권한 위임자가 권한 대리자에게 대리 인증서 정보를 보내는 반면, 제안 프로토콜은 권한 위임자가 권한 대리자에게 대리 인증서를 전달하기 위한 과정만 필요할 뿐, 권한 대리자가 권한 위임자에게 전달하는 정보는 없다. 따라서 제안 프로토콜은 권한 위임자가 자신의 권한을 위임할 경우, 권한 대리자의 키 생성을 기다릴 필요가 없이 즉각적인 권한 위임이 가능하므로 매우 효율적인 프로토콜이다. 또한 제안 프로토콜을 적용할 경우, 악의적인 권한 위임자가 권한 대리자에게 권한 위임을 위한 반복된 키 생성 요청을 함으로써 발생할 수 있는 서비스 거부 공격도 제거된다.
- 위임 추적 기능: 그림 6의 아랫부분은 O에서 A를 통하여 B에게 권한 위임을 하였을 경우 B가 획득하게 되는 대리 인증서의 사슬을 나타낸다. 기존의 X.509 대리 인증서 표준에서는 권한 위임 과정에서 생성되는 새로운 공개키 정보만이 인증서에 첨부됨으로써 추후 해당 공개키를 이용하여 A 및 B의 신원을 확인할 수 없다. 그러나 제안 프로토콜은 A와 B의 개체 공개키 p_A, p_B 정보가 대리 인증서에 포함되므로, 해당 공개키 정보를 이용하여 A, B에 대한 신원 확인이 가능하다.
- 역동성: 제안 프로토콜은 X.509 대리 인증서 있는 수준의 특성을 그대로 유지한다. 제안 프로토콜을 이용하여 위임 작업을 수행하여 특정 프로세스에게 권한 위임을 수행했다고 가정할 경우, 만약 해당 프로세스가 없어지는 상황이 발생하면 현재의 X.509 대리 인

증서에서의 위임 프로토콜과 마찬가지로 해당 프로세스가 위임 받았던 대리 인증서 및 권한 대리 개인키가 없어지며, 이것은 X.509 대리 인증서 프로토콜의 대리 인증서 파기 과정과 동일하다. 따라서 본 제안 프로토콜도 본 성질을 만족한다.

- 경량성: 계산량적인 측면에서, 본 제안 프로토콜은 기존의 프로토콜에 비해 효율적이다. 기존의 표준에서는 위임 과정시에 권한 대리자가 자신이 생성한 새로운 공개키를 권한 위임자에게 검증받기 위해 새로운 공개키에 대해 서명작업이 필요했다. 그러나, 제안 방법은 그러한 과정이 필요없다. 또한 대리 개인키를 생성하는 비용도, 제안 방법이 현재의 표준보다 효율적이다 왜냐하면 기존에 사용하던 RSA등의 알고리즘은 새로운 키를 생성하기 위한 파라미터(소수 등)를 다시 생성해야하나, 제안 방법은 기존의 파라미터를 다시 이용하여 키를 생성시키기 때문에 계산량이 매우 감소하게 된다. 네트워크 사용량을 비교하면, 제안 프로토콜에서는 표준과는 달리 권한 대리자가 권한 위임자에게 전송하는 부분이 없고, 또한 권한 위임자가 권한 대리자로 전송하는 정보의 양은 제안 프로토콜과 차이가 없기 때문에, 전체 통신 메시지 양도 줄어들게 된다. 따라서 제안 프로토콜이 기존의 표준보다 더욱 본 성질을 만족한다.
- 제한된 위임: 현재의 표준과 마찬가지로, 제안 프로토콜도 인증서를 통한 권한 위임 과정을 수행한다. 따라서 대리 인증서에 위임되는 권한에 대한 기술이 가능하며 권한 위임자는 권한 대리자에게 위임 하는 권한에 대한 제한이 가능하다.
- 반복 위임: 제안 프로토콜은 반복된 권한 위임이 가능하므로 본 기능을 만족한다.
- 개체 개인키 보호: 기존의 표준에서는 대리 개인키와 개체 개인키가 서로 아무런 관련이 없는 독립적인 관계이며, 따라서 대리 개인키가 노출됨으로써 얻을 수 있는 개체 개인키의 정보는 없다. 이에 반하여 제안 방법에서는 대리 개인키와 개체 개인키는 수학적으로 특정한 관계를 갖지만, 대리 개인키로부터 개체 개인키를 얻기 위해서는 암호학적으로 매우 어렵다고 알려진 문제를 풀어야 하며, 이것은 매우 많은 계산량을 필요로 한다. 따라서 현실적으로 대리 개인키로부터 개체 개인키를 알기는 거의 불가능하다. 따라서 제안 방법도 본 성질을 만족한다.
- 폐기의 용이성: X.509 대리 인증서 표준과 마찬가지로, 제안 프로토콜도 생성된 대리 개인키 및 대리 인증서를 없앰으로써 위임 받은 권한에 대한 폐기가 가능하다.

최종적으로, 성능과 관련한 모든 항목에 대하여 제안 방법과 현재의 대리 인증서 표준을 비교한 결과는 표 2와 같다.

표 2 제안 프로토콜과 X.509 대리 인증서 위임 프로토콜과의 성능 비교

	제안 방법	X.509 대리 인증서 표준
비대화식위임	○	×
위임추적성	○	×
역동성	◎	○
경량성	◎	○
제한된 위임	○	○
반복 위임	○	○
개체 개인키 보호	○	○
폐기의 용이성	○	○

4.2 실험 결과

본 세부절에서는 제안 방법을 이용한 권한 위임과 표준 프로토콜을 이용한 권한 위임 프로토콜의 수행 시간을 비교한다. 표준 프로토콜은 Globus Toolkit에 구현된 권한 위임 방법을 사용하였으며 사용 서명 알고리즘은 RSA이다[11]. 제안 방법의 구현을 위하여, 차재훈 및 천정희가 제안한 ID 기반 알고리즘을 사용하였다[6]. 또한 공평한 비교를 위해 RSA 알고리즘은 1024bit 길이의 공통 법수(common modulus)를 사용하며[1], ID 기반 알고리즘은 512-bit 길이의 필드 상에서 정의되는 특이 타원곡선을 이용하여 구현하였다[12]. ID 기반 알고리즘은 PBC 라이브러리[12]를 이용하여 구현하였다. 권한 위임을 수행한 환경은 Pentium IV 3.2 GHz 프로세서 및 2GB RAM을 갖는 Linux 2.6.15-1 운영체제이다. 수행 결과는 10000회 수행의 평균을 나타낸다.

수행 결과는 표 3에 기술되어 있다. 결과로부터, 제안 방법이 기존의 권한 위임 프로토콜에 비해 약 12배 정도 빠른 권한 위임이 가능함을 알 수 있다. 이러한 성능 향상의 가장 큰 이유는 RSA 알고리즘의 키 생성 알고리즘의 비효율성에 기인한다. 표 3에서 알 수 있듯이 RSA 방법을 이용한 키 생성 시간은 차재훈 및 천정희가 제안한 ID 기반 알고리즘의 키 생성 시간에 비해 대략 36배 정도 느림을 알 수 있다.

5. 결 론

본 논문에서는 현재의 X.509 대리 인증서 표준에서 제공하지 못하는 위임 추적 가능성을 제공하고 권한 위임자로부터 권한 대리자의 1회 통신만으로 위임이 가능

1) Globus Toolkit 은 512-bit 공통 법수를 갖는 RSA 알고리즘을 기본 옵션으로 사용한다[11]. 그러나 512-bit RSA 알고리즘은 1999년에 깨어졌으며[14], 현재 프로토타입 수준으로 개발된 고성능 서버를 이용할 경우 1시간 이내에 해독이 가능하다[15]. 따라서 본 알고리즘을 사용할 경우 보안상의 치명적인 문제를 발생시킬 수 있다.

표 3 제안 방법과 표준 프로토콜의 권한 위임 소요 시간 비교 (ms)

위임 프로토콜	단위 연산의 소요시간			위임을 위해 요구되는 단위 연산들	소요 시간
	K (키 생성)	S (서명생성)	V (서명검증)		
제안 방법	3.21	3.82	4.62	1K+1S+1V	11.65 ([13] 적용시)
RSA-1024 를 이용한 표준 프로토콜	117	5.52	0.18	1K+2S+2V	128.4

한 권한 위임 프로토콜을 제안하였다. 특히 광대역 통신이 빈번하게 수행되는 계산적 그리드 환경에서 그리드 도메인 간의 권한 위임자로부터 권한 대리자로의 1방향 통신만으로 권한 위임을 수행할 수 있으므로 권한 위임 시 소요되는 통신량을 줄일 뿐만 아니라, 권한 위임 과정 자체를 비 대화식으로 할 수 있으므로, 전체 그리드 환경의 성능향상에 기여한다. 또한 대리 인증서 사슬에 참여한 참여자의 신원 확인이 가능하므로, 대리 인증서 사슬에 허가되지 않은 개체가 참여할 경우, 그것을 확인할 수 있는 방법을 제공하므로, 전체 그리드 환경의 보안성 향상에 기여할 수 있다.

참 고 문 헌

[1] Globus project, http://www.globus.org [8]

[2] C. Cha and J. Cheon, "An identity-based signature from gap Diffie-Hellman groups," Proc. Public Key Cryptography - PKC 2003, Lecture Notes in Computer Science, Vol.2139, pp. 18-30, 2003. 11. [6].

[3] I. Foster, C. Kesselman, G. Tsudik and S. Tuecke, "A Security Architecture for Computational Grids," Proc. 5th ACM Conference on Computers and Communications Security, pp. 83-91, 1998. [1]

[4] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," The international journal of high performance computing applications, Vol.15, No.3, pp. 200-222, 2001. [2]

[5] F. Hess, "Efficient identity based signature schemes based on pairings," Proc. Selected Areas in Cryptography - SAC'02, Lecture Notes in Computer Science, Vol.2595, pp. 310-324, 2002. [7]

[6] B. Neuman, "Proxy-based authorization and accounting for distributed systems," Proc. 13th International Conference of Distributed Computing Systems, pp. 283-291, 1993. [5]

[7] K. Paterson, "ID-based signatures from pairings on elliptic curves," Electronics Letters, Vol.38, No.18, pp. 1025-1026, 2002. [10]

[8] L. Perlman, V. Welch, I. Foster, C. Kesselman and S. Tuecke, "Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile," RFC 3820, 2004. [3]

[9] A. Shamir, "Identity-based cryptosystems and signature schemes," Proc. of Crypto'84, Lecture Notes in Computer Science, Vol. 196, pp. 47-53, 1985. [9]

[10] V. Welch, I. Foster, C. Kesselman, O. Mulmo, L. Peralman, S. Tuecke, J. Gawor, S. Meder and F. Siebenlist, "X.509 Proxy Certificates for Dynamic Delegation," Proc. 3rd Annual PKI Workshop, 2003. [4]

[11] PBC Library (The Pairing-based Cryptography Library) http://crypto.stanford.edu/pbc [12]

[12] Y. Kawahara, T. Takagi, E. Okamoto, "Efficient Implementation of Tate Pairing on Mobile Phone using Java," International Conferences on Computational Intelligence and Security, CIS 2006, pp. 1247-1251. [13]

[13] R. P. Brent, "Recent Progress and Prospects for Integer Factorisation Algorithms," Computing and Combinatorics: 6th Annual International Conference - COCOOON 2000, LNCS, Vol.1858, pp. 3-22, 2000. [14]

[14] The Globus Alliance Website, http://www.globus.org/toolkit/downloads/4.0.5/#source [11]

[15] Teraflops Research Chip. http://techresearch.intel.com/articles/Tera-Scale/1449.htm

이 윤 호

2000년 한국과학기술원 전산학과 학사 2002년 한국과학기술원 전자전산학과 석사. 2006년 한국과학기술원 전자전산학과 박사. 2006년~2007년 한국과학기술원 정보전자연구소 박사후연구원. 2007년~현재 GTISC(GeorgiaTech Information Security Center) 방문연구원. 관심분야는 네트워크 보안, 멀티미디어 보안, 암호학

김 병 호

1990년 연세대학교 전산과학과(학사). 1992년 KAIST 전산학과(석사). 1997년 KAIST 전산학과(박사). 1997년~1998년 포스데이타 과장. 1998년~2005년 브레인21 대표이사. 2007년~현재 경성대학교 전임강사. 관심분야는 센서 네트워크