악의적 지니 프록시로부터 비밀 정보 보호를 위한 암호학적 모델

양 종 필[†] · 이 경 현[††]

요 약

가까운 미래에 사람들은 언제 어디서나 컴퓨팅 서비스를 받기 위하여, 다양한 형태의 이기종 네트워크들로 접속할 것이며, 네트워크의 이기종성으로 인하여 다양한 형태의 운영·보안 프로토콜이 혼재할 것이다. 이동 장치들은 다양한 이기종 네트워크들에 접속하기 위해서 미들웨어에 의존함으로써, 각 네트워크에서 사용되는 프로토콜에 대한 구체적인 지식 없이 원하는 서비스를 받을 수 있을 것이다. 만약 이동 장치와 서비스간의 안전한 채널이 요구될 경우, 이동 장치에서 동작하는 미들웨어는 안전한 채널 설정을 위한 암호 프로토콜을 성공적으로 수행하기 위하여 이동 장치내의 개인키(Private key)에 접근을 해야만 한다. 본 논문에서는 이동 장치에서 동작하는 악의적 미들웨어로부터 개인키를 보호하기 위한 암호학적 모델을 제안한다. 제안되는 모델에서는 이동 장치에 개인키를 저장하지 않은 채 암호 프로토콜을 성공적으로 수행하기 위한 인증 데이터(즉, 전자 서명문) 생성이 가능하다.

키워드 : 미들웨어 보안, 유비쿼터스 컴퓨팅

A Cryptographic Model to Protect Private Information against Malicious Proxy in Jini

Jong-Phil Yang[†] · Kyung Hyune Rhee[††]

ABSTRACT

In the near future, people will wish to access many kinds of heterogeneous networks to use their services anytime and anywhere. Owing to the heterogeneity of networks, there must be many kinds of protocols to guarantee secure services. The mobile device can depend on a middleware for accessing services in the heterogeneous networks and the middleware helps the mobile device to communicate with services without knowing concrete protocols. If a secure channel is necessary, the middleware may access a private key in the mobile device to perform a security protocol. In this paper, we focus on the security of a private key in the mobile device against malicious middlewares. To do so, we introduce two models for a user to protect his/her private key against malicious middlewares by generating authentication data(e.g., digital signatures) without keeping the private key in the mobile device.

Key Words : Middleware Security, Ubiquitous Computing

1. 서 론

지난 몇 년간 이동 통신의 급격한 증가와 더불어, 이동 장치들은 유비쿼터스화 되고 있으며, 데스크 탑, PDA, 이동 전화기 등과 같은 장치들에 대한 구분이 점차적으로 희미해져 가고 있다. 가까운 미래에 사람들은 언제 어디서나 컴퓨팅 서비스를 받기 위하여, 다양한 형태의 이기종 네트워크들로 접속할 것이며, 네트워크의 이기종성으로 인한 다양한 서비스들의 혼재는 많은 형태의 운영·보안 프로토콜들이 존재하게 됨을 의미한다. 따라서, 이동 장치들이 다양한 이기종 네트워크들에 접속하기 위해서, 각 네트워크에서 사용되는 프로토콜에 대한 구체적인 지식 없이 원하는 서비스를 받을 수 있어야 할 것이다.

선 마이크로 시스템즈(Sun Microsystems)에서 개발한 지니(Jini)는 컴퓨팅 장치가 미들웨어(middleware)를 통하여 이기종 네트워크들 내의 다양한 형태의 자원들로부터 서비스들을 제공받을 수 있는 시스템이다. 어떤 사용자가 어떤 자원의 서비스를 받고자 할 때, 그는 그 자원이 발행한 실행 코드(executable code)를 다운받기만 하면 된다. 그 사용자는 서비스 사용을 위한 구체적인 프로토콜에 대한 지식이나 장치 드라이버의 설치 없이, 다운로드한 실행 코드를 통해서 그 자원과 서비스를 위한 통신을 성공적으로 수행하게

※ 본 연구는 정보통신부에서 주관하는 IT분야 해외유학 지원사업의 결과물임.
† 준 회 원 : 일본 류슈 시스템 정보기술 연구소(ISIT) 연구원
†† 종신회원 : 부경대학교 전자컴퓨터정보통신공학부 교수(교신저자)
논문접수 : 2005년 7월 20일, 심사완료 : 2005년 11월 30일

된다. 모든 통신이 다운로드한 실행 코드를 거쳐가기 때문에, 사용자의 개인키(private key) 보호에 대한 문제점이 발생할 수 있다. 예를 들어, 다운로드한 코드가 인증과 기밀성을 만족하는 보안 채널을 설정하기 위해서, 이동 장치 내에 저장되어 있는 사용자의 개인키에 접근해야 하는 상황을 고려해 보자. 사용자는 다운로드한 코드가 개인키를 다른 목적으로 사용하거나 제 삼자에게 노출시키지 않는다는 보장을 받지 못하기 때문에, 다운로드한 코드가 사용자의 개인키로 접근하는 것을 원하지 않을 것이다. 비록 그 다운로드한 코드는 발행자에 의해서 전자 서명되어 있더라도, 전자 서명의 검증은 그 코드의 실행의 올바름을 보장해 주지는 못한다. 현재, 인터넷 환경에서 전자 상거래(e-commerce)와 전자 뱅킹(e-banking)에서 안전한 거래를 위하여 서명된 엑티브 엑스(ActiveX) 코드들이 사용되고 있으며, 엑티브 엑스 코드에 대한 전자 서명의 검증이 성공한 경우 아무런 제약없이 사용자의 로컬 머신에 존재하는 사용자의 개인키에 직접적으로 접근한다. 이와 같은 코드들은 신뢰할 만한 개체인 은행 또는 쇼핑몰에 의해서 전자 서명이 되었기 때문에, 사용자들의 개인키로 접근이 허용되고 있는 실정이다.

"커피숍에서의 문서 출력 서비스" 또는 "소극장에서의 티켓 발행 서비스"와 같이, 사용자들은 다양한 이기종 네트워크 환경에 존재하는 매우 다양한 자원이 제공하는 서비스들을 받기 위하여 다양한 실행 코드들을 다운로드하게 될 것이며, 그 자원들은 은행과 같이 신뢰할 만한 개체가 아닐 수 있기 때문에, 다운로드한 코드들에 대한 실행의 올바름을 확신하기 어렵다. 또한, 사용자가 자신이 소유한 다수의 이동 장치들을 통하여 이기종 네트워크에 접속을 하길 원할 경우, 그 사용자는 다수의 개인키 사본을 생성하여 각각의 이동 장치들에게 저장해야만 한다. 물론, 사용자는 개인키를 스마트 카드에 저장할 수 있으나, 그 사용자는 안전한 접속을 위해서 이동 장치와 스마트 카드 모두를 항상 소지하고 있어야 한다. 또한, 스마트 카드는 이기종 네트워크에서 요구되는 서비스들를 위한 다양한 암호학적 알고리즘들을 제공하기 어렵다.

본 논문에서는 사용자의 이동 장치들이 악의적인 실행 코드로부터 개인키들의 노출에 대한 위협 없이 인증 데이터(즉, 전자 서명문)를 생성할 수 있는 암호학적 모델을 소개하고자 한다. 생성된 인증 데이터는 이동 장치들이 인증, 보안 채널의 설정 및 서비스 사용에 따른 과금을 위한 증거 등과 같이, 다양한 형태의 운영·보안 프로토콜들을 위하여 사용될 것이다. 2장에서는 관련 연구로서 지니 보안을 위한 사전의 연구를 살펴보고, 3장에서 제안 모델의 구조와 응용 시나리오를 소개한다. 그리고, 4장에서는 제안 모델의 구체적 실현을 위한 두 가지의 암호학적 기법을 통한 구체적인 모델을 설계한다. 5장에서는 제안 모델에 대한 고려 사항과 특징을 살펴보고, 마지막으로 6장에서 결론을 맺는다.

2. 관련 연구

선 마이크로 시스템즈에서 개발한 지니(Jini)는 이동 장치들이 수동 설치 및 조정 없이 ad hoc 커뮤니티를 생성할 수 있다. 각 장치들은 다른 장치들에게 서비스를 제공하기 위하여, 서비스 이용에 필요한 디바이스 드라이버 및 소프트웨어를 "프록시(proxy)"라는 다운로드 가능한 실행 코드로서 제공한다. 따라서, 지니는 클라이언트 장치와 서비스를 제공하는 장치들 사이에 어떠한 특정 통신 프로토콜들을 사전에 명시하지 않고, 단지 동적인 자바 클래스 로딩을 통해서 모든 통신 및 동작에 대한 프로토콜이 상호간에 정의될 수 있다. 지니 환경에서 어떤 서비스들을 제공하는 자원을 "서비스"라고 하고, 그 서비스에 의해서 발행된 실행 가능한 코드를 "프록시"라 하며, 프록시는 클라이언트 장치와 서비스 사이의 통신을 처리하기 위한 자바 객체이다. 서비스는 "룩업 서비스(lookup service)"라는 또 다른 서비스에 자신의 프록시를 등록시킴으로써, 다른 장치들에게 자신의 기능들을 제공할 수 있는 준비상태가 된다.

어떤 서비스를 사용하고자 하는 클라이언트 장치는 먼저 룩업 서비스를 사용하기 위하여, 룩업 서비스의 프록시를 다운로드 하기 위한 디스커버리(Discovery) 프로시저를 수행한다. 클라이언트는 원하는 서비스에 대하여 룩업 서비스에게 질의한다. 물론, 질의 또한 룩업 서비스의 프록시를 통해서 수행하게 된다. 만약 클라이언트가 원하는 서비스를 발견하면, 룩업 서비스로부터 원하는 서비스를 위한 프록시를 다운로드한다. 이 순간부터, 클라이언트 장치와 서비스간의 모든 통신은 룩업 서비스의 개입 없이 다운로드한 프록시를 통하여 이루어진다[9].

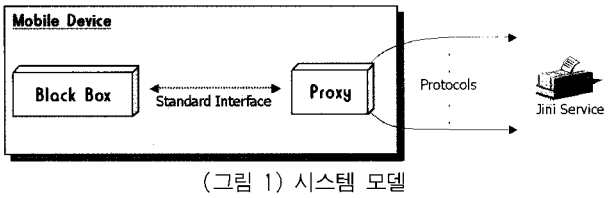
지니는 자바 언어를 기반으로 하기 때문에, 지니 보안 기술은 자바 보안 기술과 높은 연관성을 가지고 있다. 따라서, 지니 보안을 위한 연구는 자바 보안 기술에 중심을 둔 기술들이 제안되고 있다. 다운로드한 코드에 대한 권한(permission) 부여는 "어디에서 코드가 다운로드되었는가?"와 "누가 그 코드를 전자 서명하였나?"에 의해서 이뤄지고 있으며, JAAS(Java Authentication and Authorization Service) 또한 대한 권한 부여를 위한 확장으로 사용되고 있다[6]. P. Hasselmeyer 등은 [8]에서 룩업 서비스의 보안성 향상을 위한 연구에 초점을 두었다. P. Eronen 등은 지니 보안을 위한 요구사항들을 소개하였으며, 지니 서비스에 접속하는 클라이언트를 보호하기 위한 솔루션을 제안하였다[4], [5]. P. Eronen 등의 솔루션은 SPKI(Simple Public Key Infrastructure)을 기반하여 다운로드된 프록시에 대한 접근 제어를 수행하기 위하여, 지니 보안 매니저(Jini security manager)라는 새로운 형태의 보안 매니저를 부가적으로 필요로 하고 있다. F. Sommers는 JSK 2.0 [14]에서 지니의 새로운 보안 특징(인증, 무결성, 동적인 권한 부여 등)들을 소개하였지만, JSK 2.0의 새로운 보안 특징들은 클라이언트 장치내의 개인키 보호를 위한 명확한 솔루션은 제공하지 못하고 있다.

현존하는 지니와 자바를 위한 보안 구조는 악의적인 프록시로부터 클라이언트 장치 내의 개인키를 보호하기 위한 솔루션을 제공하지 못하고 있다. 기존의 보안 구조에서는 다운로드한 코드에 대한 클라이언트 장치내의 접근 제어를 위

한 규칙은 "다운로드한 코드의 발행자(예, 전자 서명자 또는 네트워크 주소 등)", "전자 서명의 유효성"과 "발행자의 인증서에 대한 검증"을 통해서 다운로드 수행 시에 정의된다. 따라서, 올바른 발행자로부터 올바른 전자 서명이 이루어진 악의적 프록시의 경우, 안전한 지니 서비스를 수행하기 위하여 클라이언트 장치내의 개인키로 접근할 수 있는 권한을 다운로드 시에 소유하게 될 것이다. 따라서, 악의적인 프록시로부터의 개인키 보호는 기존의 지니와 자바를 위한 보안 구조를 통해서는 이루어 질 수 없다. 악의적인 프록시로부터 클라이언트 장치 내에 존재하는 개인키를 보호하기 위한 솔류션은 암호학적 모델로서 새롭게 설계함으로써 제공될 수 있다. 본 논문에서는 지니 서비스를 사용하고자 하는 이동 장치내의 개인키를 새로운 보안 관리자의 추가 또는 자바 가상 머신(Java Virtual Machine)의 변경 없이 안전하게 보호할 수 있는 암호학적 모델의 설계를 목적으로 한다. 따라서, 지니 서비스와 룩업 서비스 자체의 보안성에 대한 연구는 본 논문의 주요 연구 대상이 아니다. 또한, 본 논문에서는 스마트 카드와 같이 개인키를 안전하게 사용하기 위한 추가적인 물리적 장치를 고려하지 않는다.

3. 시스템 모델

본 논문에서의 클라이언트 장치는 랩탑, PDA, 핸드폰과 같은 가까운 미래에 컴퓨팅 파워가 더욱 증가된 이동 장치를 대상으로 한다. 사용자의 이동 장치가 이기종 네트워트내의 어떤 서비스에 접속하기를 원할 때, 그 이동 장치는 대상 이기종 네트워트를 위한 룩업 서비스를 통하여 서비스를 위한 프록시를 다운로드한다. (그림 1)은 본 논문에서 제안하는 모델의 기본 구조를 보여주고 있다.



(그림 1) 시스템 모델

프록시는 인증 데이터(즉, 전자 서명문)를 단지 표준 인터페이스(Standard Interface)를 통해서만 블랙 박스(Black Box)로부터 획득할 수 있다. 여기서, 블랙 박스는 동작 방식에 따라서 다양한 형태의 자바 클래스로서 구현될 수 있으며, 자바 클래스로 정의된 표준 인터페이스 또한 블랙 박스의 구현에 의존적으로 구현된다. 따라서, 본 논문에서 제안되는 모델의 블랙 박스는 프록시에게 서명 오라클(signing oracle)로서 동작하게 된다. 본 논문에서는 다음과 같은 사항을 가정한다.

- 지니 룩업 서비스는 반드시 안전한 것은 아니다. 즉, 온라인 신뢰 개체로서 가정하지 않는다.
- 이동 장치(명백히, 자바 가상 머신내의 보안 관리자)는 다운로드한 프록시에게 동적으로 권한을 부여할 수 있다.
- 이동 장치는 이기종 네트워크내의 어떤 서비스를 받기 위해서는 다운로드한 프록시를 거쳐야 한다.
- 이동 장치와 프록시 모두가 표준 인터페이스를 알아야 한다. 따라서, 표준 인터페이스를 통해서만 프록시와 블랙 박스는 통신이 가능하다. 즉, 프록시는 블랙 박스로부터 인증 데이터를 얻기 위해서 표준 인터페이스를 사용한다.
- 이동 장치는 표준 인터테이스를 구현한 자바 클래스를 소유하고 있으며, 이동 장치는 필요에 따라 그 자바 클래스를 구현한 객체(object)를 프록시에게 제공할 수 있다.
- 이동 장치의 사용자는 자신의 개인키를 비교적 안전한 장소 (즉, 이동 장치가 아닌 장비. 예, 데스크탑)에 보관하고 있다.

본 논문에서 제안된 모델에 대한 실질적인 활용 방법을 문서 출력 시나리오를 통해서 살펴본다.

[Step 0] 사전에 서비스(즉, 프린터)는 자신의 프록시를 개인키로 전자 서명한 후, 룩업 서비스에 등록을 한다.
[Step 1] 지니 문서 출력 서비스를 사용하고자 하는 사용자는 룩업 서비스에 접속하여, 문서 출력을 제공하는 서비스에 대한 질의를 수행한다. 사용자는 곧 문서 출력을 제공하는 서비스의 리스트를 이동 장치의 화면을 통해서 볼 수 있게 된다.
[Step 2] 사용자는 자신의 문서를 출력하기 위하여, 리스트된 서비스들 중 하나를 선택하다. 곧, 사용자의 이동 장치에 직렬화된 프록시 객체(serialized proxy object)가 전송되고, 대응되는 바이트 코드(bytecode)가 다운로드된다.
[Step 3] 이동 장치는 프록시에 대한 전자 서명의 유효성을 검증한다. 만약 성공적이면, 이동 장치는 프록시가 수행하는 보안 또는 동작 프로토콜의 수행에 요구되는 최소한의 권한만을 부여한다. 여기서, 프록시는 표준 인터페이스를 통한 블랙 박스로의 접근에 대한 권한을 부여받아야만 한다.
[Step 4] 프록시는 표준 인터페이스를 위한 직렬화된 객체와 대응되는 바이트 코드를 획득한다. 그 후, 프록시는 문서 출력을 위해서 서비스에 연결을 시도한다. 프록시는 문서 출력을 위한 프로토콜들을 알고 있으며, 프록시가 프로토콜 수행 중에 인증 데이터가 필요하게 되면, 획득된 바이트 코드를 통해서 블랙 박스로부터 인증 데이터를 얻을 수 있다.

위의 문서 출력 시나리오에서, 블랙 박스로부터 생성된 인증 데이터는 수행되는 "보안 프로토콜의 입력 파라메터" 또는 "서비스 사용에 대한 과금을 위한 정보"로서 사용될 수 있다. 본 논문에서 제안되는 모델의 보안 목표는 아래와 같다.

[Goal 1] 이기종 네트워크내의 서비스로부터 다운로드한 프록시는 암호학적 연산을 위해서 이동 장치내의 개인키로 직

접적인 접근 없이, 요구되는 인증 데이터(즉, 전자 서명문)를 획득할 수 있어야 한다.

[Goal 2] 이기종 네트워크내의 서비스에 안전하게 접근하기 위해서, 악의적인 프록시로부터 개인키 자체의 노출에 대한 위협이 없어야 한다.

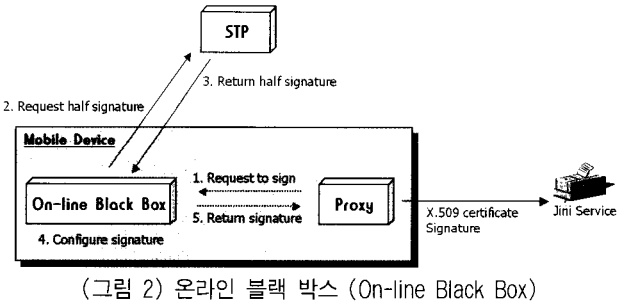
[Goal 3] 이동 장치의 장비 또는 성능을 고려하여, 인증 데이터를 생성하기 위한 다양한 방법들이 적용 가능해야 한다. 즉, 이동 장치가 소유한 통신 설비 및 컴퓨팅 능력을 고려하여 적절한 블랙 박스가 적용될 수 있어야 한다.

4. 블랙 박스의 설계

본 장에서는 사용자의 인증 데이터(즉, 전자 서명문)을 생성하는 블랙 박스의 두 가지 구현 모델인 "온라인 블랙 박스(on-line black box)"와 "오프라인 블랙 박스(off-line black box)"를 소개한다. 온라인 블랙 박스는 사용자의 이동 장치가 광대역 네트워크(즉, 인터넷)에 언제나 접속 가능한 경우 사용할 수 있으며, 오프라인 블랙 박스는 광대역 네트워크에 제한적으로 접속 가능한 경우 사용될 수 있다.

4.1 온라인 블랙 박스 (On-line Black Box : ONBB)

온라인 블랙 박스(ONBB)의 구현을 위하여 [3], [16]에서 제안된 양자간 전자 서명 기법을 암호학적 프리미티브로서 사용한다. 사용자의 개인키는 두 부분으로 나눠져서 한 부분은 사용자가 소유하고, 나머지 부분은 부분 신뢰된 개체(Semi-Trusted Party : STP)[2]가 소유하게 된다. (그림 2)는 ONBB의 전체적인 동작 방식을 보여주고 있다.



(그림 2) 온라인 블랙 박스 (On-line Black Box)

사용자의 이동 장치는 STP와 협력을 통해서만 인증 데이터(즉, 전자 서명문)을 생성할 수 있다. X를 사용자의 개인키, Y를 공개키로 두자. 그러면, 인증 데이터 생성을 위한 구체적인 절차는 아래와 같다.

[초기 설정]

[Step 1] 사용자는 자신만이 알고 있는 패스워드 (δ)를 정의하고, 의사 랜덤 함수 PRF를 사용하여 랜덤 비트 시퀀스 $\Delta = PRF(\delta)$를 획득한다. 그리고, 사용자는 $X_{STP} = X \circ \Delta$를 계산한다. 여기서, X_{STP}는 기반 전자 서명 기법의 키 생성 규칙에 따르도록 해야 하며, $\circ$ 연산자 또한 기반 전자 서명 기법에 의존한다.

[Step 2] 사용자는 X_{STP}를 STP에게 안전하게 등록한다.

[인증 데이터 생성]

프록시가 수행하고 있는 보안·운영 프로토콜의 성공적인 수행을 위해서 어떤 메시지 m에 대한 전자 서명을 표준 인터페이스를 통해서 요청하면, ONBB는 다음을 수행한다.

[Step 1] ONBB는 m을 STP에게 전송한다. STP는 부분 서명문 $S_{STP} = g_1(X_{STP}, m)$을 계산하여 ONBB에게 리턴한다. 여기서, g_1 알고리즘은 기반 전자 서명 기법에 의존한다.

[Step 2] STP로부터 부분 서명문을 수신하면, ONBB는 $S_{ONBB} = g_2(\Delta, m)$을 계산하고, m에 대한 전자 서명문 $S = g_3(S_{STP}, S_{ONBB})$를 유도한다. 여기서, g_2와 g_3 또한 기반 알고리즘에 의존한다. ONBB는 S_{ONBB}를 계산하기 위하여, 사용자에게 패스워드(δ) 입력을 요청할 것이다. 최종적으로, ONBB는 S를 프록시에게 리턴한다.

(그림 3)은 [3]에서 소개된 mRSA를 통한 ONBB를 구현한 예를 보여주고 있다. 여기서, RSA와 같이, 사용자는 두 큰 소수 p와 q를 통하여 $N = p \times q$을 계산한다. 그리고, $\gcd(e, \phi(N)) = 1$와 $d \times e = 1 (\mathrm{mod} \phi(N))$를 만족하는 공개키 (e, N)과 개인키 (d, N)을 계산하여 소유하고 있는 것으로 가정한다. 그리고, $H(\cdot)$는 RSA 전자 서명을 위한 PKCS#1 또는 OAEP와 같은 패딩함수이다.

[초기 설정]

[Step 1] 사용자는 $d_{STP} = d - \Delta$를 계산한다. 여기서, Δ는 사용자 패스워드로부터 유도된 랜덤 비트 시퀀스이다.

[Step 2] 사용자는 d_{STP}를 STP에게 안전하게 등록한다.

[인증 데이터 생성]

[Step 1] ONBB는 $H(m)$을 STP에게 전송한다. STP는 $S_{STP} = H(m)^{d_{STP}} \mathrm{mod} N$을 계산하고, ONBB에게 리턴한다.

[Step 2] ONBB는 Δ를 계산하기 위하여 사용자에게 패스워드 입력을 요청한다. 사용자가 패스워드를 입력하면, ONBB는 $S_{ONBB} = H(m)^{\Delta} \mathrm{mod} N$을 계산하고, $S = S_{ONBB} \times S_{STP} \mathrm{mod} N$을 계산한다. 그리고, ONBB는 S를 프록시에게 리턴한다.

(그림 3) mRSA 기반의 ONBB

ONBB로부터 수신한 S를 통하여, 프록시는 보안·운영 프로토콜을 성공적으로 완료할 수 있다. 또한, ONBB를 사용함으로써, 사용자는 이동 장치에 개인키를 저장하지 않은 채 전자 서명문을 생성할 수 있다. 하지만, ONBB는 다음과 같은 제약 사항들을 가지고 있다.

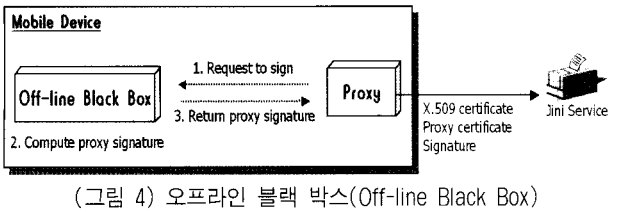
2) 본 논문에서는 부분 신뢰 개체(Semi-trusted party)에 대한 개념 소개는 생략한다. 좀 더 상세한 정보는 [3]을 참고하길 바란다.

- 사용자의 개인키의 부분 정보가 단일 STP에 저장되어 있기 때문에, 네트워크 단절로 인한 일시적 서비스 거부, STP의 심각한 오류로 인한 영속적인 서비스 거부와 같은 문제점이 발생할 수 있다. 따라서, STP는 항상 온라인 및 보안을 유지해야만 한다.
- 이동 장치는 인증 데이터를 생성하기 위해서 항상 STP에 접속을 해야만 하기 때문에, 이동 장치는 IEEE 802.11g와 같은 광대역 네트워크에 접속할 수 있는 장치를 소유해야만 한다.
- STP로부터 ONBB로 전송되는 부분 서명문(S_{STP})에 대한 기밀성이 보장되지 않는 환경에서, 공격자는 S와 S_{STP}를 통한 오프라인 사전 공격에 대한 위협이 존재한다. 따라서, 사용자는 주의 깊게 패스워드를 선정해야만 한다.
- 기본적으로 STP는 ONBB로 부터의 부분 서명문 계산 요청의 정당성 유무를 판단할 수 없다. 그러므로, 부순 서명문 계산 요청에 대한 인증 프로토콜이 필요할 것이다.

다음절에서 소개된 ONBB의 제약 사항들을 개선하기 위한 새로운 형태의 블랙 박스를 소개할 것이다. 이동 장치는 자체적으로 인증 데이터를 생성함으로써 더 이상 온라인에 대한 요구가 없다.

4.2 오프라인 블랙 박스 (Off-line Black Box : OFBB)

오프라인 블랙 박스(OFBB)의 구현을 위해서, [1]에서 제안된 대리 전자 서명 기법(proxy signature scheme)의 일반적인 구조법인 "인증서에 의한 위임(delegation by certificate)"를 사용한다. [13]에서 소개된 대리 인증서(proxy certificate) 또한 "인증서에 의한 위임"을 사용한 실례로 간주할 수 있다. OFBB에서는 사용자의 개인키는 대리 전자 서명 기법의 원 서명키(original signing key)의 역할을 하게 되며, 사용자는 이동 장치에게 서명 능력을 위임하기 위하여 대리 서명키(proxy signing key)를 생성한다. 그러므로, 이동 장치는 원 서명키(즉, 사용자의 개인키)가 아닌 대리 서명키를 사용하여 인증 데이터를 생성한다. (그림 4)는 OFBB의 전체적인 동작 방식을 보여주고 있다.



(그림 4) 오프라인 블랙 박스(Off-line Black Box)

4.1절과 같이 X를 사용자의 개인키(즉, 원 서명키), Y를 공개키로 두자. 그러면, 인증 데이터를 생성하기 위한 절차는 아래와 같다.

[초기 설정]

[Step 1] 사용자는 기반 전자 서명 기법에 따라서 이동 장치(M)를 위한 키 쌍을 생성한다; 대리 서명키(X_M) 그리고 대리 공개키(Y_M). 그리고, 사용자는 허가 정보(warrant)와 대리 인증서를 생성한다. 여기서, 대리 인증서는 허가 정보, 사용자의 신원, 이동 장치의 식별자, Y_M 등을 포함하고 있으며, X를 사용하여 전자 서명된다.

[Step 2] 사용자는 이동 장치 내에 대리 인증서와 X_M을 안전하게 저장한다.

[인증 데이터 생성]

프록시가 어떤 메시지 m에 대한 전자 서명을 표준 인터페이스를 통해서 OFBB에게 요청하면, OFBB는 다음을 수행한다.

[Step 1] OFBB는 대리 전자 서명문 $S = g_1(X_M, m)$을 계산한다. 여기서, g_1 알고리즘은 기반 전자 서명 기법에 의존한다.

[Step 2] OFBB는 S를 프록시에게 리턴한다.

프록시는 S를 이동 장치와 지니 서비스간의 보안·운영 프로토콜 수행을 성공적으로 수행하도록 한다. 또한, S가 대리 전자 서명문이기 때문에, 프록시는 S와 함께 대리 인증서를 지니 서비스에게 전송해야만 한다. OFBB에서 사용된 "인증서에 의한 위임"의 경우, 원 서명자(즉, 사용자)가 모든 대리 전자 서명문의 생성에 대한 책임을 가지기 때문에, 대리 전자 서명 기법의 "대리 보호(proxy protected), 강한 위조 불가(strong unforgeability), 강한 부인 방지(strong undeniability), 오용 방지(prevention of misuse)"와 같은 일반적인 보안 요구 사항을 만족시킬 필요가 없다[10], [11]. OFBB는 다음과 같은 장점을 가진다.

- OFBB에는 부분 신뢰 개체(Semi-trusted party)가 불필요하다.
- 이동 장치는 인증 데이터를 생성하기 위하여 광대역 네트워크 접속이 불필요하다.
- 이동 장치가 침해되었을 경우, 대리 서명키는 노출될 위협이 존재하지만 사용자의 개인키는 노출되지 않는다.

(그림 5)는 [15]에서 소개된 RSA 기반의 (t, n)-임계 대리 전자 서명 기법(threshold proxy signature scheme)의 임계 전자 서명이 아닌 형태로 변형한 것이며, 사용자의 키 생성 규칙 및 키 쌍은 (그림 3)의 전제 조건과 동일하다. OFBB의 경우, 사용자는 각 이동 장치들을 위해서 주기적으로(예, 하루 또는 2-3일 주기로서) 대리 서명키와 대리 인증서를 발행해야만 한다. 그리고, 대리 전자 서명문의 검증자(즉, 지니 서비스)는 ONBB와 비교하여 대리 인증서 검증을 위한 부가적인 자원 소모를 감수해야만 한다.

[초기 설정]

[Step 1] 사용자는 이동 장치(M)에게 서명 능력을 위임하기 위하여 RSA 키 쌍과 허가 정보(warrant)를 생성한다; 대리 서명키(d_M, N_M), 대리 공개키(e_M, N_M) 그리고 허가 정보(W_M). 그리고, 사용자는 이동 장치를 위한 대리 인증서 $K = H(W_M, DN, M, e_M)^d \bmod N$를 계산한다. 여기서, DN은 사용자의 공개키 (e, N)에 대한 X.509 인증서내의 DN(Distinguished Name)이다.

[Step 2] 사용자는 이동 장치 내에 K와 (d_M, N_M)을 안전하게 저장한다.

[인증 데이터 생성]

[Step 1] OFBB는 대리 전자 서명문 $S = H(m)^{d_M} \bmod N_M$을 계산한다.

[Step 2] OFBB는 S를 프록시에게 리턴한다.

(그림 5) Delegation by certificate 기반의 OFBB

5. 고려 사항 및 고찰

본 장에서는 소개되었던 ONBB와 OFBB에서의 실질적 시스템 배치에서 발생할 수 있는 두 가지 문제점들과 그에 대한 해결책들을 고려한다.

5.1 ONBB에서 부분 전자 서명문 계산 요청에 대한 인증

사용자의 개인키는 이동 장치내에 저장되어 있지 않기 때문에, STP가 부분 서명문 계산 요청의 유효성을 판단하기 위하여 공개키 암호 기반의 인증 프로토콜은 사용할 수 없다. 그러므로, 본 논문에서는 S/Key와 같은 OTP(One Time Password) 기반의 인증 프로토콜이 좋은 해결책이 될 것으로 판단된다[7].

ONBB에서 사용자는 패스워드를 Δ를 유도하기 위해서 사용하기 때문에, 동일 패스워드는 S/Key의 해쉬 체인 내에서 현재 반복수만큼 계산된 해쉬값(OTP)을 계산하기 위해서 사용될 수 있다. 따라서, ONBB가 STP에서 S_{STP}를 요청할 때, 현재 반복수에 대응되는 해쉬값(OTP)을 함께 전송하면, STP는 수신한 해쉬값(OTP)의 유효성 검증을 통해서 부분 전자 서명문 계산 요청의 정당성을 확인할 수 있게 된다.

5.2 OFBB에서 대리 전자 서명문 검증을 위한 속도 향상

일반적인 전자 서명문이 사용되는 ONBB에서는 CA의 X.509 인증서와 사용자의 X.509 인증서로 구성된 인증서 체인(certificate chain)이 존재하게 되며, 전자 서명문을 검증하기 이전에 인증서 체인의 유효성이 사전에 검증되어야 한다. 그러나, OFBB는 대리 인증서가 부가적으로 인증서 체인에 추가되므로, 대리 전자 서명문의 검증자(지니 서비스)는 ONBB에 비하여 인증서에 대한 전자 서명 검증을 위한 부가적인 연산을 부담해야만 한다. 따라서, 검증자의 인증서 체인 검증에 대한 부담을 줄이기 위해서 [2], [12]에서 제안된 전자 서명문 결합(signature aggregation) 기술이 적용될 수 있다. 비록 검증자가 인증서 체인내의 모든 링크에 대해서 알고 있어야 하지만, 결합된 전자 서명문(aggregated signature)은 인증서 체인에 대한 압축이 가능하다. 즉, n명의 서로 다른 사용자로부터의 n개의 서로 다른 메시지에 대한 n개의 전자 서명문이 주어졌을 때, 모든 전자 서명문을 단일 전자 서명문을 결합시키는 것이 가능하다. 따라서, 검증자는 결합된 단일 전자 서명문의 단일 검증을 통해서, n명의 사용자들이 n개의 메시지를 전자 서명했다는 것을 확인 가능하다.

사용자가 OFBB 초기 설정의 (Step 1)을 수행할 때, 그 사용자는 자신의 X.509 인증서와 대리 인증서를 결합하여 "결합된 인증서(aggregated certificate)"를 생성하고, 대리 인증서를 대신하여 이동 장치에 저장시킨다. 따라서, 검증자(지니 서비스)는 대리 전자 서명문의 검증에 필요한 인증서 체인에 대한 검증 속도를 가속화 가능하다. (그림 5)에 대한 전자 서명문 결합 기술의 실질적 구현을 위해서, [12]에서 소개된 "제한이 없는 법 연산을 가지는 순차 결합 전자 서명(sequential aggregation signature under unrestricted moduli)"를 사용할 수 있다.

이동 장치의 설비에 의존하여, 지니 프록시와 블랙 박스 사이의 표준 인터페이스는 다양한 형태의 자바 클래스로 구현할 수 있다. 그리고, 표준 인터페이스의 실질적인 구현에 따라서, 이동 장치는 프록시가 표준 인터페이스에 대한 자바 클래스의 객체를 통하여 블랙 박스에 접근하도록 동적인 권한 부여를 해야한다.

본 논문에서 제안된 두 가지의 암호학적 모델에 대한 구체적인 장·단점들은 이미 4장에서 상세히 논하였다. 다음은 제안 모델의 전반적인 주요 장점들에 대해서 요약한다.

- **사용자 비밀 정보에 대한 보호:** ONBB와 OFBB를 통하여, 사용자는 개인키를 이동 장치에 저장하지 않고, 필요한 인증 데이터를 생성할 수 있다. 전자 서명이 성공적으로 검증된 악의적인 프록시는 "동작 프로토콜의 수행에 요구되는 최소한의 권한"과 "표준 인터페이스를 통한 블랙 박스로의 접근에 대한 권한"만을 소유할 수 있다. 따라서, 사용자의 비밀 정보(즉, ONBB에서는 사용자 패스워드, OFBB에서는 대리 서명키)로의 직접적인 접근에 대한 권한은 주어지지 않으며, 이는 자바 보안 구조에서 권한에 대한 자바 보안 관리자의 접근 제어를 통해서 안전성이 보장될 수 있음을 의미한다.
- **자바 가상 머신에 대한 변경이 불필요:** [4], [5]와 달리, 제안 모델을 위해서 기존의 자바 가상 머신에 존재하는 보안 관리자 이외의 추가적 보안 관리자가 불필요하다. 또한, 모든 가정 사항과 제안 모델의 구조는 현재의 자바 가상 머신 환경 하에서 구현 가능하다.
- **인증 데이터 생성을 위한 사전 초기화:** [4], [5]의 경우, 이동 장치는 다운로드한 각 프록시에 대한 전자 서명 검증이 성공적으로 수행되면, 그 다운로드한 프록시에 대한 집

근 제어와 시스템 보호를 위하여, 즉시적인 키 쌍 및 SPKI 인증서 생성을 해야한다. 하지만, 위와 같은 키 쌍의 생성 및 인증서의 생성은 컴퓨팅 파워가 취약한 이동 장치에게는 매우 큰 자원 낭비를 초래한다. 하지만, 제안 모델은 위와 같은 키 쌍과 인증서 생성 절차가 불필요하다.

- **개인키 저장의 불필요 :** [4], [5]의 경우, 부가적 보안 관리자인 지니 보안 관리자가 프록시에게 인증서를 발행하기 위하여, 이동 장치에 개인키가 저장되어 있어야 한다. 이미 서술한 바와 같이, 제안 모델에서는 이동 장치에 개인키 자체를 저장할 필요 없이 프록시가 필요로 하는 인증 데이터 계산이 가능하다. 여기서, 악의적인 프록시가 이동 장치내의 모든 비밀 정보에 대한 접근 권한을 소유하게 되는 매우 강력한 악의적인 프록시를 가정해 보자. [4], [5]의 경우는 사용자의 개인키, ONBB는 사용자의 패스워드, OFBB는 대리 서명키가 강력한 악의적인 프록시에게 노출될 것이다. 하지만, 본 논문에서 제안된 블랙 박스 모델의 비밀 정보(즉, 패스워드 및 대리 서명키)의 유효 기간을 짧게 부여함으로써, 위와 같은 악의적 프록시로부터의 피해를 최소화 할 수 있다. 즉, 패스워드와 대리 서명키에 대한 짧은 유효기간을 부여한다는 것은 시스템의 복잡도를 증가시키지는 않으나, 사용자는 안전하게 수행되는 각 블랙 박스 모델의 "초기 설정" 단계를 빈번히 수행해야 하는 불편함을 가지게 된다.

6. 결 론

본 논문에서는 악의적인 지니 프록시로부터 사용자의 개인키를 보호하기 위한 암호학적 모델을 제안하였으며, 현재의 자바와 지니 보안 모델을 기초로 하여, 안전한 지니 시스템을 개발하기 위한 중요 기술로서 적용될 수 있으리라 기대한다. 비록, 현재의 이동 장치들의 성능이 공개키 암호 알고리즘 계산을 효율적으로 수행하기에 어려움이 있으나, 가까운 미래에는 제한된 컴퓨팅 파워는 더 이상 제약이 되지 않을 것으로 판단된다. 본 논문에서 제안된 모델을 위한 자바 클래스의 설계와 실질적 구현은 향후 과제로 남겨둔다.

참 고 문 헌

[1] A. Boldyreva, A. Palacio, B. Warinschi, "Secure proxy signature schemes for delegation of signing rights," Cryptography ePrint Archive, Report 2003/096 http://eprint.iacr.org/(2003).

[2] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "A Survey of Two Signature Aggregation Techniques," In Crypto Bytes Vol.6, No.2 (2003).

[3] D. Boneh, G. Tsudik, G.M. Wong, "A method for fast revocation of public key certificates and security capabilities," The 10th USENIX Security Symposium, pp.297~308 (2001).

[4] P. Eronen, J. Lehtinen, J. Zitting, P. Nikander, "Extending Jini with Decentralized Trust Management," The 3rd IEEE Conference on Open Architectures and Network Programming, pp.25~29 (2000).

[5] P. Eronen, P. Nikander, "Decentralized Jini Security," The Network and Distributed System Security Symposium (NDSS), pp.161~172 (2001).

[6] J. Garms, D. Somerfield , "Professional Java Security," Wrox Press Ltd (2001).

[7] N. Haller, "The S/KEY One-Time Password System," The ISOC Symposium on Network and Distributed System Security (1994).

[8] P. Hasselmeyer, R. Kehr, M. VoB, "Trade-offs in a Secure Jini Service Architecture," The 3rd IFIP/GI International Conference on Trends towards a Universal Service Market(USM), pp.190~201 (2000).

[9] W. Keith Edwards , "Core Jini : 2nd Edition", Prentice Hall (2001).

[10] J .Y. Lee, J.H. Cheon, S.J. Kim, "An Analysis of Proxy Signatures : Is a Secure Channel Necessary?," CT-RSA 2003, LNCS 2612, pp.68~79 (2003).

[11] B. Lee, H. Kim, K. Kim, "Strong Proxy Signature and its Applications," SCIS2001, Vol.2/2, pp.603~608 (2001).

[12] A. Lysyanskaya, S. Micali, L. Reyzin, H. Shacham, "Sequential Aggregate Signatures from Trapdoor Permutations," Cryptography ePrint Archive, Report 2003/091, http://eprint.iacr.org/ (2003).

[13] A. Romao, M. M. da Silva, "Secure Mobile Agent Digital Signatures with Proxy Certificates," E-Commerce Agents : Marketplace Solutions, Security Issues, and Supply and Demand, LNCS 2033, pp.206~220 (2001).

[14] F. Sommers, "Jini Starter Kit 2.0 tightens Jini's security framework," May, 9 (2003) http://www.javaworld.com/javaworld/jw-05-2003/jw-0509-jiniology.html.

[15] J. Yang, K. H. Rhee, "Collaborative Admission Control in the Office," The Third International Workshop for Applied Public Key Infrastructure(IWAP), pp.49~59 (2004).

[16] J. Yang, S. U. Shin, K. H. Rhee, "A Simplified Approach

to User Controllable Threshold Signatures", The IEEE Conference on Electronic Commerce(CEC), pp.273~280 (2004).

양 종 필

e-mail : mahasuli@naver.com

1999년 부경대학교 전자계산학과(학사)

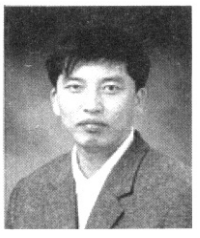
2001년 부경대학교 대학원 전자계산학과 (석사)

2005년 부경대학교 대학원 전자계산학과 (박사)

2005년~현재 일본 큐슈대학교 시스템정보과학부 객원연구원

일본 큐슈 시스템 정보기술 연구소(ISIT) 연구원

관심분야 : 비밀 분산, 공개키 기반 구조, 인터넷 보안 프로토콜 등



이 경 현

e-mail : khrhee@pknu.ac.kr

1982년 경북대학교 수학교육과(이학사)

1985년 한국과학기술원 응용수학과(이학석사)

1992년 한국과학기술원 수학과(이학박사)

1982년~1993년 한국전자통신연구소 선임연구원

1993년~현재 부경대학교 전자컴퓨터정보통신공학부 교수

1995년~1996년 호주 애들레이드대학 방문교수

2001년~2002년 미국 UCI 방문교수

2002년~2003년 국제간 정부기구 CPSC 교학부장

2004년~현재 한국정보보호학회 학술이사

2003년~현재 한국멀티미디어학회 재무이사

관심분야 : 암호이론, 멀티미디어 정보보호, 네트워크 보안, 암호 프로토콜