공 학 석 사 학 위 논 문

모바일 환경을 위한 개선된 DAA 인증 프로토콜

2008년 2월

부 경 대 학 교 대 학 원

컴 퓨 터 공 학 과

이 기 열

공 학 석 사 학 위 논 문

모바일 환경을 위한 개선된 DAA 인증 프로토콜

지도교수 정 목 동

이 논문을 공학석사 학위논문으로 제출함

2008년 2월

부 경 대 학 교 대 학 원

컴 퓨 터 공 학 과

이 기 열

이기열의 공학석사
학위논문을 인준함

2008년 2월 26일

주　　심　공학박사　이 경 현　인

위　　원　공학박사　신 상 욱　인

위　　원　공학박사　정 목 동　인

목 차

그 림 목 차

표 목 차

모바일 환경을 위한 개선된

DAA 인증 프로토콜

이 기 열

부 경 대 학 교 대 학 원 컴 퓨 터 공 학 과

요약

TCG (Trusted Computing Group) 에서 제안하고 있는 TPM (Trusted Platform Module) 과 DAA 프로토콜은 미래의 신뢰 컴퓨팅 기술을 제공하기 위한 새로운 보안 메커니즘이다. DAA는 Chaum과 Heyst에 의해 제안된 그룹 서명과 Camenisch와 Lysyanskaya에 의해 제안된 서명 스킴을 기반으로 하는 새로운 개념의 원격 인증 프로토콜이다. 그러나 DAA 프로토콜은 많은 계산량을 요구하고, 구현의 어려움으로 인해 기술 확산에 어려움을 가지고 있다. 특히 국내에서는 TPM 및 DAA 프로토콜에 대한 연구가 많이 진행되지 않고 있어 미래의 신뢰 컴퓨팅 기술에 대한 연구가 시급한 실정이다.

이러한 문제들을 해결하기 위해서 본 논문에서는 신뢰 컴퓨팅 모듈인 TPM과 신뢰 컴퓨팅 구축을 위한 DAA 프로토콜을 소개하고, DAA 프로토콜을 수학적으로 분석하여 DAA 프로토콜이 가지는 문제점을 제시한

다. 또한 이들 문제점에 대한 해결책과 계산량에 대한 문제 해결을 위해서 모바일 환경에 적합한 DAA 프로토콜을 제안할 것이다.

새롭게 제안되는 모바일 환경을 위한 DAA 프로토콜은 영지식 증명, Camenisch-Lysyanskaya 스킴의 단일 메시지 서명 기법과 Camenisch-Stadler의 서명 기법의 장점을 활용하여 기존의 DAA에 비해 적은 계산량을 요구하고 프라이버시 문제를 해결한다. 제안하는 보안 스킴은 효율적인 DAA 프로토콜 제공을 통해 PC 환경보다 적은 자원과 제한된 환경을 가진 모바일 디바이스를 위한 신뢰 컴퓨팅 및 원격 인증 기술을 제공할 수 있을 것이다.

An Improved DAA Authentication Protocol for the Mobile Environment

Ki Yeal Lee

Department of Computer Engineering, Graduate School,

Pukyong National University

Abstract

TCG(Trusted Computing Group) is providing a new security mechanism using TPM and DAA protocol, supporting future trusted computing technology. DAA is a new concept of remote authentication using Group signature, and Camenisch-Lysyanskaya signature scheme. But due to huge computation overhead and implementation difficulty, proliferation of this technology is still uncertain. Especially, domestic research is hardly going on. Thus, it is urgent for us to take a action for future research regarding trusted computing technology.

In order to solve a problem stated above, this paper introduces TPM, the trusted computing module, and DAA protocol which builds trusted

computing, and analyzes it mathematically to find out the problem of DAA protocol. Also, we propose new DAA protocol that is suitable for mobile environment in order to solve the current problem and computation overhead.

Since new protocol that is proper for mobile environment takes advantage of single message signature from Camenisch & Lysyanskaya scheme, It provides more efficient protocol and low computation, and less privacy problem comparing to current DAA. Security scheme proposed in this paper can provide trusted computing environment and remote authentication technology for a mobile device which has lower and limited resource than PC.

1. 서론

1.1 연구의 필요성 및 현황

최근 네트워크와 컴퓨팅 기술의 발전으로 인터넷 사용량이 급격히 증가하였고 이로 인해 웹상에서 인증기법의 필요성이 증가하고 있다. 인증(Authentication)이란 자신의 신원을 다른 사람에게 알리고 증명하는 것을 말한다. 현실 세계에서 사용되는 신분증이 이런 역할을 하는 대표적인 예이다. PKI(Public Key Infrastructure)는 이러한 인증기법 중 가장 널리 사용되는 방법 중 하나이다[29].

그러나 이러한 서비스들은 웹상에서 안전한 인증을 수행하기 위해 사용자의 개인 정보를 필요로 하고, 더 높은 보안 강도를 위해 더 많은 사용자의 개인 정보를 필요로 하게 된다. 이는 사용자의 신원, 즉 개인 정보가 웹상에 유포될 수 있는 가능성을 높이고, 인터넷 서비스를 이용하는 사용자들에게 불안감을 높이는 원인이 되었다. 이로 인해 높은 보안 기능은 사용자에 대한 프라이버시를 침해하는 경향을 띄고 있다.

또한 컴퓨팅 환경의 확산에 따라 컴퓨팅 디바이스에 대한 외부의 위협이 날로 증가하고 있다. 현재의 보안 기술은 소프트웨어의 보안 기술이 대부분이다. 하지만 현존하는 소프트웨어 기반의 보안 방식의 한계가 점차 밝혀지면서 하드웨어 기반의 보안 기능을 제공하고자 하는 방안이 연구되고 있다[22. 23].

앞의 두 문제점을 해결하기 위해서 Intel, IBM, Sun, 삼성전자와 같은 대형 IT 기업들은 TCG(Trusted Computing Group)[21]를 결성하였다. TCG는 이종 컴퓨팅 플랫폼에서 컴퓨팅 환경의 보안을 강화시키려고 만

들어진 비영리 기관으로서 신뢰할 수 있는 컴퓨팅 플랫폼의 표준을 개발하기 위하여 다양한 스펙을 제안하고 있다[21, 22, 25, 26, 27].

특히 TCG는 TPM (Trusted Platform Module) 이라는 디바이스를 제안함으로 PC, 임베디드 플랫폼, 모바일 환경에 대한 보안 기능을 강화하고자 노력하고 있다. TPM은 TCG에 의해 제정된 산업 표준 규격을 기초로 한 보안 칩으로 구현된 하드웨어 보안 디바이스이다. 이는 플랫폼에 장착되어 보안과 관련된 정보를 저장하고 사용하며, 하드웨어 무결성을 보장하고자 하는 목적을 가지고 있다.

1.2 연구의 내용

TPM과 상호 연동되는 많은 기기들에 대한 체계적인 인증 및 보안 관리를 위해서는 TPM의 기본적인 보안 및 인증 기술이 필요하다. 이러한 요구를 충족시키기 위해서 TPM은 Privacy CA (TPM 1.1버전) [21, 22]와 DAA (TPM 1.2버전) [6]을 도입하고 있다.

Privacy CA 방식은 기존의 PKI가 가지는 아이디어를 따라 만들어진 개념으로 플랫폼을 인증(Attestation)하기 위해서는 Privacy CA를 통해야 한다. 이는 PKI가 가지는 단점과 같이 Privacy CA에게 인증에 필요한 중요 기능이 집중되어 CA에 대한 부하가 커지고 CA가 공격당할 가능성이 크다는 단점을 지니고 있다. 또한 TPM은 Privacy CA를 통해서 인증할 때 프라이버시 보호를 위해서 AIK(Attestation Identity Key)를 매번 만들어야 한다는 단점을 가지고 있다.

이러한 이유로 TCG는 스펙 1.2버전을 제안하면서 Privacy CA를 없애고 DAA라는 새로운 프로토콜을 적용하였다. 그룹서명의 전문가인 Camenisch, Erikbell, Chen의 아이디어를 통해 제안된 DAA 프로토콜은

그룹서명을 기반으로 하고 있으며, 영지식 증명(Zero knowledge proof)을 사용하여 사용자(TPM)의 익명성을 보장하면서 사용자를 인증할 수 있는 기능을 제공한다.

DAA의 장점은 TPM을 소유한 사용자가 다른 TPM과의 통신을 하고자 원하거나 원격 서비스를 사용하고자 할 때 자신의 신분을 밝히지 않고 안정적인 보안 서비스를 제공받을 수 있게 해준다는 것이다.

1.3 논문의 중요성 및 구성

DAA는 복잡성에 대한 문제로 인해 많은 계산 량과 구현의 어려움을 가지고 있고, 이러한 문제들은 DAA의 상용화를 늦추는 원인이 되고 있다. 문제점 해결을 위해 IBM, Intel를 중심으로 경량화 된 DAA 프로토콜에 대한 연구가 진행 중이다[9, 10, 17, 30]. 세계 대형 IT 기업들의 이러한 노력에도 불구하고 국내에서는 TPM 및 DAA 프로토콜에 대한 연구가 많이 진행되고 있지 않아 미래의 보안 기술에 대한 연구가 시급한 실정이다.

따라서 본 논문에서는 DAA 프로토콜에 대한 분석과 DAA가 가지는 문제점을 제시할 것이다. 또한 이러한 문제점을 해결하기 위한 수정된 DAA 프로토콜 및 아이디어를 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 TPM 및 DAA 기술에 대해서 기술하고, DAA와 관련된 수학적인 지식을 소개할 것이다. 3장에서는 DAA 분석을 통해서 가질 수 있는 문제점을 분석하고, 문제점 해결을 위한 방안과 새로운 DAA 프로토콜을 제시한다. 4장에서는 DAA 프로토콜 및 제안된 프로토콜의 구현 및 평가를 제시한다. 끝으로 5장에서는 결론 및 향후 연구에 대해서 살펴본다.

2. 관련 연구

이 장에서는 DAA (Direct Anonymous Attestation) 프로토콜의 기초가 되는 TPM (Trusted Platform Module), 그룹서명 기술 [5], Camenisch와 Lysyanskaya의 서명 스킴 [1, 12], DAA 프로토콜의 수학적 기반 등에 대해서 살펴봄으로써 DAA 프로토콜을 분석하고자한다.

2.1 TCG와 TPM

2.1.1. TCG (Trusted Computing Group)

TCG는 이종 컴퓨팅 플랫폼에서 컴퓨팅 환경의 보안을 강화시키고자 하는 목적으로 설립된 비영리 산업 표준화 기관이다. 그림 1과 같이 몇 개의 중심 기업들이 대표로 구성되어 정책을 결정하면서 여러 개의 워킹 그룹과 기술 위원회를 두고 있다. TCG는 컴퓨터 플랫폼 내부로의 “Root of Trust"를 결합하기 위한 목적을 가지고 있다. TCG는 소프트웨어 적인 솔루션만으로는 하드웨어 컴퓨팅 플랫폼에 대한 보안 기능을 더 이상 보장할 수 없다고 생각하여 신뢰 컴퓨팅 플랫폼 (Trusted Computing Platform) 을 기반으로 하는 하드웨어와 운영체제의 성능 향상을 위해서 스펙을 개발하고 있다.

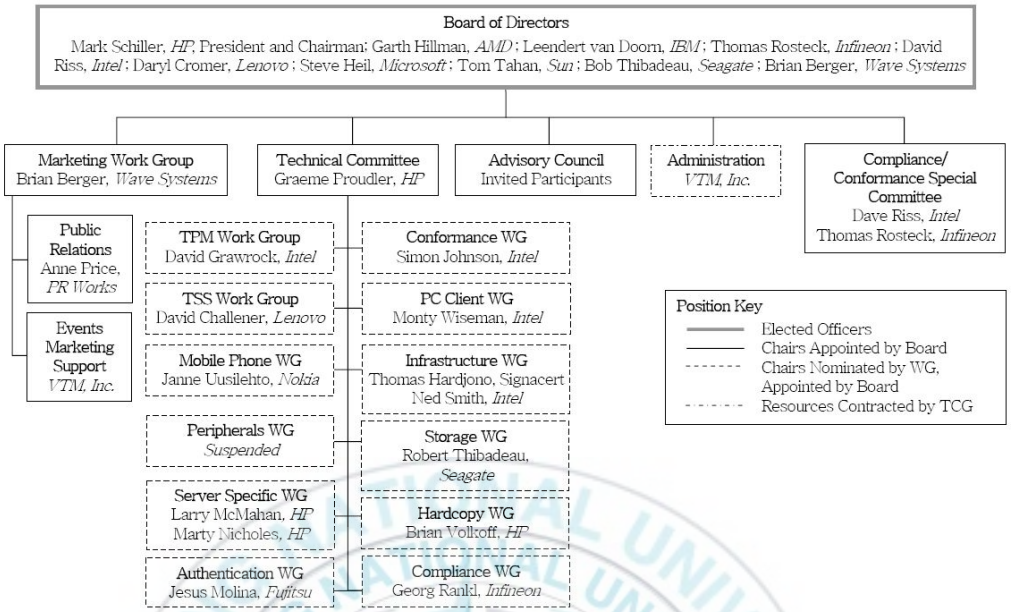

그림 1. TCG의 구성 [26]

2.1.2. TCG의 아키텍처

(1) TPM

TPM은 TCG에 의해 제안된 산업 표준 규격을 기초로 한 보안 칩으로 마이크로 컨트롤러, 암호 엔진, 표준 입출력 인터페이스, 안전한 메모리를 갖추고 공개키, 디지털 인증서, 민감 데이터 보호등과 같은 기능을 제공한다. 또한 TPM은 디바이스에 임베디드 되어야 하므로 낮은 전력, 고성능의 프로세스 설계를 요구한다.

(가) TPM의 기능

- 키 보호 : EK(Endorsement Key), AIK(Attestation Identity Key)와 같은 다양한 키들을 TPM내에 저장한다.
- 시스템 인증 : DAA 또는 Privacy CA를 통하여 플랫폼에 대한 인증을 처리한다.
- RNG : 안전한 키 생성을 위해 하드웨어-기반의 난수를 생성한다.
- 안전한 저장 : 민감한 데이터를 저장하는 PCR(Platform Configuration Register)에 SHA-1 해시 알고리즘을 이용하여 저장한다.

(나) TPM의 특징

TPM은 플랫폼의 신뢰(Trust of Platform)를 제공하기 위하여 다음과 같은 특징을 가진다.

- TPM은 외부 소프트웨어 공격이나 물리적인 공격으로부터 저장된 정보를 안전하게 보호한다.
- 암호모듈과 보호 저장소(Protected storage)를 내장하여 플랫폼의 integrity를 측정한 결과를 PCR(Platform Configuration Register)로 불리는 특정 레지스터에 저장한다.
- 암호 모듈과 난수 발생기 등을 사용하여 플랫폼의 인증과 Root of Trust를 이용하여 무결성(Integrity)을 검증하는 인증(Attestation)을 수행한다.

(다) TPM의 인증 방식

TCG에서는 플랫폼이 외부의 공격이나 내부의 다른 요인에 의해 변경이나 손상되지 않았음을 나타내는 무결성을 신뢰의 기본으로 한다. 따라서 신뢰 컴퓨팅에서는 플랫폼과 사용자의 데이터를 보호하기 위해서 기존의 인증(Authentication)을 사용하고, 플랫폼의 무결성을 검증하기 위해서 인증(Attestation)을 이용한다. TCG에서 제안하는 TPM의 인증(Attestation)의 방식은 다음과 같이 3가지로 나뉜다.

- TPM에 의한 인증 (Attestation by TPM) : AIK (Attestation Identity Key)를 이용해서 TPM 내부데이터를 디지털 서명한다. AIK는 Privacy CA나 DAA 프로토콜을 통해 획득 가능하다
- 플랫폼에 대한 인증 (Attestation to the platform) : 플랫폼에 대한 무결성 측정의 결과를 믿을 수 있다고 증명한다. 플랫폼이 들고 있는 인증서등을 이용해서 수행하고 AIK를 발급할 때 사용된다.
- 플랫폼 인증 (Attestation of the platform) : 플랫폼의 무결성 측정에 대해 증명한다. TPM의 AIK를 이용한 PCR의 디지털 서명에 의해 이루어진다.

본 논문은 플랫폼 인증 (Attestation of the platform)을 위한 DAA 프로토콜을 분석하고, 문제점을 개선한 DAA 프로토콜을 제안할 것이다.

(라) TPM의 컴포넌트

다음은 TPM의 컴포넌트의 구조의 기능을 나타낸다.

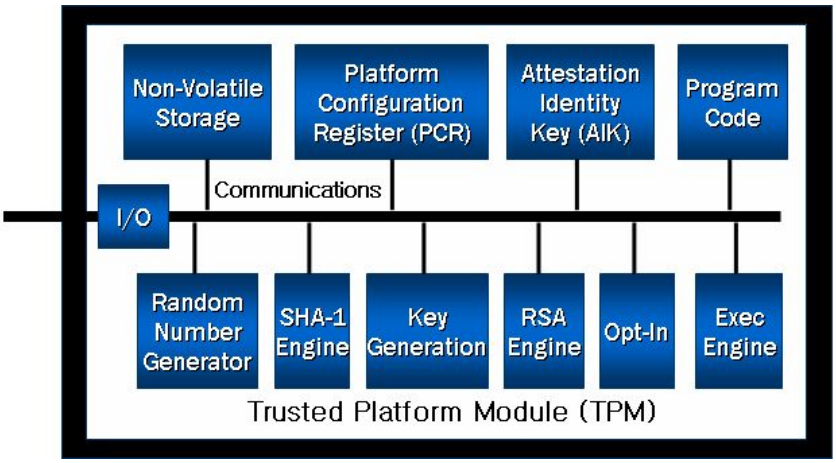

그림 2. TPM 컴포넌트의 구조 [21, 22]

- I/O : 통신 버스상의 정보의 흐름을 관리하고, 내부 및 외부 버스상의 통신에 적합한 프로토콜을 인코딩 및 디코딩한다. 또한 메시지를 적절한 컴포넌트로 라우팅하며, 접근 제어에 필요한 정책을 수행한다.
- Non-volatile Storage : EK (Endorsement Key) 및 SRK (Storage Root Key), 사용자 인증 데이터, 영구 플래그를 저장한다.
- Platform Configuration Register (PCR) : 부팅 과정에서 현재 상태와 PCR 레지스터에 저장된 레퍼런스와 비교 하여 값이 같다면 신뢰한다. 공격이나 바이러스 등으로 인한 변조는 이러한 과정을 통해 감지되어 사용자는 이에 대하여 적절한 조치를 취할 수 있다.
- Attestation Identity Keys (AIK) : 생성된 데이터를 서명하는데 사용하는 특별한 키로서 서비스 제공자에게 플랫폼 인증을 제공하기 위해 사용한다.
- Program Code : 플랫폼 디바이스를 측정하기 위한 펌웨어를 포함한다.
- Random Number Generator (RNG) : TPM에 의해서 보호되는 비밀키의 생성과 키의 임시 생성, 그리고 범용적인 목적을 위한 하드웨어 기

반 난수 생성기이다.

- SHA-1 엔진 : 서명 계산, Key BLOB 생성 등, 범용의 목적으로 사용되는 해시 함수로 데이터의 무결성을 위해서 사용된다.
- RSA Key Generation : AIK와 SRK를 생성하기 위해 이용된다.
- RSA Engine : 서명키로 서명, 저장키로 암호화 / 복호화 및 EK로 복호화를 수행한다.
- Opt-In : 물리적 현존 상태를 결정하고 불능의 조작을 보증하는 것을 결정하는 로직으로 수요자의 요구에 따른 TPM의 기능에 대한 TCG 정책을 이행한다.
- Execution Engine : 프로그램 코드를 실행 하고 TPM의 초기화와 측정 작업을 수행한다.

(2) TSS(TC Software Stack)

다른 하드웨어의 구성요소와 마찬가지로 TPM은 특정 드라이버와 서비스 제공 인터페이스를 필요로 하는데 이를 제공하기 위해서 신뢰 플랫폼 지원 서비스인 TSS[23, 26]가 제공되고 있다. TSS는 애플리케이션에게 TPM 기능에 대한 단일 엔트리 포인트를 제공하면서 동기화된 엑세스 기능 및 리소스 관리 기능을 제공한다. TSS는 이러한 기능을 제공하기 위해서 TPM은 TPM 함수들을 제공하는 보안 API를 구성하고 있다. 다음 그림3은 TSS의 구조를 나타내고 있다.

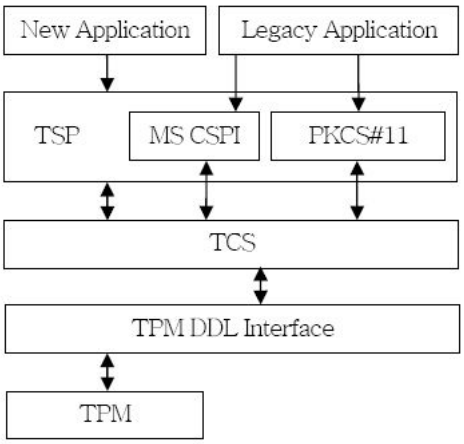

그림 3. TSS의 구조[26]

TPM은 애플리케이션과 통신을 위해서 TPM DDL 인터페이스를 통해서 TCS와 통신을 한다. TCS는 일반 플랫폼 서비스에 대한 인터페이스를 제공하고 하나의 플랫폼에 여러 개의 TSP가 존재하더라도 TCS는 모든 TSP에서 정상적으로 작동을 한다. TCS는 플랫폼과 관련된 키의 저장 및 PCR 레지스터와의 엑세스 관리와 같은 기능을 수행한다. TSP는 TPM에 대한 C 언어 인터페이스를 제공한다. 또한 TSP에서는 보호 기능을 효율적으로 사용하기 위하여 MS가 주도하는 MS 보안 서비스 프로바이더(MS Cryptographic Service Provider)와 RSA 사의 PKCS#11을 제공하고 있다.

2.2 그룹 서명

2.2.1 그룹 서명의 개념

그룹 서명[5]은 1991년 David Chaum과 Eugene van Heyst가 처음으로

제시한 멀티 전자서명의 한 방법으로 그룹의 멤버들은 자신을 드러내지 않고 익명성을 유지하면서 서명을 제시하는 방법이다. 그룹서명은 다음과 같은 3가지 특징을 가지며 이를 꼭 만족해야 한다.

- 그룹의 멤버만이 서명 가능하다.
- 서명을 받은 사람은 서명을 통해 서명자가 그룹의 멤버라는 사실은 알 수 있으나, 정확히 누구인지는 알 수 없다.
- 특별한 경우, 그룹 관리자는 서명을 개봉해서 누가 서명을 했는지 찾아내는 것이 가능하다.

그룹 서명은 어떠한 그룹을 이루는 개개인에게 익명성을 제공하면서 전자 서명 기능을 수행하게 해주는 이점이 있다. 하지만 분쟁이 발생할 경우 분쟁을 해결하기 위하여 누구의 서명인지를 밝혀내는 익명성 파기(Open) 기능을 요구하고 있다. 그룹 서명은 시작(Setup), 준비(Join), 서명(Sign), 검증(Verify), 익명성 파기(Open) 이 다섯 단계를 통하여 전자 서명의 생성과 검증이 가능하고, 문제가 발생하면 문제를 해결 할 수 있다.

- 시작 (Setup) : 그룹의 공개키와 그룹관리자의 관리키를 생성한다.
- 준비 (Join) : 새 그룹의 소속 원과 그룹 관리자간의 프로토콜로 소속원의 비밀키와 소속원임을 증명하는 인증서를 생성한다.
- 서명 (Sign) : 메시지 m과 구성원의 비밀키를 통해서 m 에 대한 서명을 생성한다.
- 검증 (Verify) : 메시지 m과 서명, 그룹의 공개키를 이용하여 서명의 유효성을 검증한다.
- 익명성 파기 (Open) : 서명을 개봉해서 생성한 소속원의 신분 및 증

거를 확인한다.

2.2.2 그룹 서명의 요구 조건

그룹서명은 그룹에 속한 소속 원들의 개개인 프라이버시를 보호하고 소속 원들이 원하는 전자서명 기능을 제공해야 한다. 이러한 기능을 제공하기 위해서 그룹서명은 다음과 같은 5가지 요구조건을 만족해야 한다.

- 건전성 (Soundness)과 완전성 (Completeness) : 그룹 멤버에 의해 서명된 올바른 서명은 항상 검증이 가능해야 하고, 올바르지 않은 서명은 항상 실패해야 한다.
- 위조 불가 (Unforgeable) : 그룹 멤버만이 올바른 서명을 생성할 수 있어야 하며, 위조가 불가능해야 한다.
- 서명자 숨기기 (Signer Ambiguous) : 메시지와 그 메시지의 서명이 주어지더라도 관리자의 비밀키가 없으면 서명자를 알아 낼 수 없다.
- 서명 연결 불가 (Uniteability) : 두 개의 메시지와 그것의 서명이 주어지더라도 그것이 같은 서명자가 서명했는지를 알아낼 수 없다.
- 공모 불가 (No Framing) : 그룹의 멤버와 관리자가 몰래 공모하더라도 다른 멤버의 서명을 위조할 수 없다.
- 생성되지 않은 서명 처리 불가 (Unforgeable tracing verification) : 그룹 관리자는 서명자가 생성하지 않은 서명에 대하여 부정하고 처벌할 수 없다.

2.2.3 그룹 서명 기반의 종류

(1) RSA 가정 기반의 그룹 서명

RSA 가정 (Assumption) 기반[13]의 그룹 서명은 임의로 생성된 RSA 모듈러 n, 지수 r, 임의의 $z \in Z_n^*$에서 $y^r = z$인 y를 찾는 것이 어렵다는 점을 이용한 그룹서명이다. 이 방법은 1991년 Chaum과 Heyst가 최초로 제시한 그룹 서명에서 이용되었다. 하지만 이러한 방법은 그룹 멤버가 늘어나면 서명의 길이도 늘어나고, 익명성 파기(Open)의 경우 익명성을 보장할 수 없다는 단점을 지니고 있다.

(2) Strong RSA 가정 기반의 그룹 서명

Strong RSA 가정 (Assumption)[13, 16] 기반의 그룹 서명은 임의로 생성된 RSA 모듈러 n, 임의의 $z \in Z_n^*$에서 $y^r = z$인 y와 r>1을 찾는 것이 어렵다는 점을 이용한 그룹서명이다. 이 방법은 공모 공격에 대한 안전성을 높였다는 장점을 가지고 있지만, 그룹 멤버가 늘어나면 서명의 길이도 길어진다는 단점을 가지고 있다. 대표적인 그룹서명으로는 Camenisch-Lysyanskaya서명 알고리즘이 존재한다.

(3) ID 기반의 그룹 서명

ID 기반의 그룹 서명 방식[3, 24]은 사용자 개인의 고유 ID를 키 생성 단계에서 도입하는 방식이다. 이는 1997년 박상준의 ID-based 그룹 서명이라는 논문에서 처음 제시되었다. 이는 그룹 서명에 따라 그룹의 공개키와 서명이 비례하여 커진다는 단점과 익명성 파기(Open)가 될 경우 익명성이 보장 되지 않는다는 단점을 지니고 있다.

2.2.4 Chaum과 Heyst의 그룹 서명

Daivd Chaum과 Eugene van Heyst가 제안한 그룹 서명[5] 논문은 다음의 가정을 기반으로 한다.

- 가정 1: 각 구성원들이 RSA의 해를 계산하는 것은 불가능하다.
- 가정 2: 각 구성원들은 모듈러 n을 알고 있더라도 n이 충분히 크면 이산대수를 계산하는 것은 불가능 하다.

이 논문은 4가지의 그룹 서명 스킴을 제시하고 있다. 이 스킴들은 위의 가정 1, 2를 기반으로 안전한 그룹 서명 기법을 제공하나 그룹에 참여하는 사람이 증가하면 선형적으로 서명의 길이가 늘어난다는 단점을 지니고 있고, 효율성에도 많은 문제점을 지니고 있다.

2.2.5 그룹 서명의 동향

David Chaum과 Eugene van Heyst가 최초로 그룹 서명을 제시하면서 문제점으로 제기한 것이 서명의 크기가 그룹 멤버의 수에 비례해서 증가한다는 것이다. 이는 그룹의 멤버 수에 제한을 두게 되므로 큰 문제점이 된다. 하지만 2000년에 Ateniese[8]등이 서명의 크기가 고정된 그룹 서명 방법을 제안하면서 이 문제를 해결할 수 있게 되었고 그룹서명에 대한 관심이 더욱 증가하였다.

또한 2004년 Dan Boneh[24]등이 타원 곡선 상의 Bilinear Pairing과 Strong RSA assumption을 이용해서 서명의 크기를 혁신적으로 줄이는데 성공했다. 이전의 서명의 크기가 1000byte가 넘는데 비하여 Bilinear

Pairing을 이용하면 200byte 미만으로 서명의 크기를 줄이는 것이 가능하면서 안전성에는 전혀 영향을 미치지 않는 것이 가능하다. 게다가 Strong Diffie-Hellman assumption을 이용하여 랜덤 오라클 모델을 이용하지 않고서도 안전성을 증명하는 것이 가능해졌다. 이 또한 그룹 서명이 처음 제시 되었을 때 제기된 문제점인 Fiat-Shamir[7] 방법을 사용하지 않고도 그룹서명을 생성할 수 있는 가라는 문제점을 해결하는 것이 가능하게 하였다.

이렇게 처음에 David Chaum과 Eugene van Heyst가 제기한 문제점들은 시간이 가면서 차차 해결이 되어 가고 있다. 또한 현재 전 세계가 하나의 네트워크로 연결되어 가는 시점에서 사용자의 개인정보의 보호가 무엇보다도 중요해지는 이때에 그룹 서명에 대한 관심은 점점 높아져 가고 있다.

개인정보의 보호라는 분야 외에도 그룹서명의 익명성을 이용한 다양한 프로젝트가 진행되고 있다. 예를 들어 미국 교통부에서 연구 중인 자동차 안전 통신 시스템이 있다. 이는 자동차에 단거리 통신기를 장착하여 주변의 자동차들과 통신을 하여 사고의 위험을 줄이는 것을 연구하는 것이다. 그 외에도 익명성을 이용하는 다양한 프로젝트에 그룹서명을 접목시키려는 노력이 계속되고 있다[14, 24, 28].

2.3 Camenisch-Lysyanskaya 서명 스킴

전자 서명 스킴은 암호학에서의 중심적인 기술로서, Diffie-Hellman 공개 키 암호화 기술과 함께 개발되었고, Rivest, Shamir, Adleman에 의해 처음 실용화 되었다. 특정 메시지의 인증을 위하여 서명을 생성하거나 특정한 목적에 사용되는 애플리케이션을 위한 메시지 블록을 생성하는데 사

용된다. 예를 들어 전자 화폐, 그룹 서명, 익명 인증서 시스템(anonymous credential system)같은 애플리케이션을 구현하기 위해 전자 서명 스킴이 사용된다.

여기서 제시된 Camenisch-Lysyanskaya(이하 CL) [1, 12] 서명 스킴은 Strong RSA 가정 (assumption)이라는 수학적 가정의 어려움에 기반을 두고 좀 더 효율적인 익명 인증서 시스템 (Anonymous credential system)을 구현하는데 알맞은 서명 스킴이다. CL 서명 스킴은 DAA 프로토콜을 기반으로 하고 있으며, 영지식 기반으로 사용자의 비밀정보 노출 없이 그룹서명이 가능한 방식이다.

2.3.1 CL 서명 스킴의 특징

- CL 서명 스킴은 다른 애플리케이션을 위해 블록을 만들 수 있는 서명 스킴을 만들고자 제시되었다.
- 익명 인증서 시스템 (Anonymous credential system)을 설계하기 위해 사용된다.
- 시스템을 사용하기 위한 인증서를 교부 받는 과정을 익명성을 유지하면서 수행한다.
- 영지식 증명을 사용하여 비밀 정보를 추가적인 정보를 제공하지 않고 증명해야 한다.
- CL 서명 스킴은 Strong RSA assumption에 기반을 두고 있고, 현재 알려진 증명 가능한 서명 스킴 가운데서 Cramer-Shoup 서명 스킴 다음으로 효율성 면에서 우수함을 보인다.

2.3.2 CL 서명 스킴의 구성 요소

CL 서명 스킴은 다음과 같이 구성되어있다.

- 키 생성 (key generation) 알고리즘: 공개키 PK와 비밀키 SK를 생성한다.
- 서명 (Signing) 알고리즘: 메시지 m과 비밀키 SK를 입력으로 받아 서명을 생성한다.
- 검증 (Verification) 알고리즘: 어떤 스트링이 공개키 PK로 암호화된 메시지 m에 대한 서명인지를 검증한다.

2.3.3 CL 서명 스킴의 흐름 분석

CL 서명의 흐름 분석을 하기 전에, CL 서명 스킴이 가지는 전제에 대해서 설명하고자 한다. CL 서명 스킴은 다음과 같은 전제를 기반으로 그룹 서명 기법을 제공한다.

- 안전한 소수 (safe prime) : p′가 소수일 때, p = 2p‘ + 1인 p는 안전한 소수라 하고, 대응되는 수 p는 Sophie Germain 소수로 알려져 있다.
- 특별한 RSA 모듈러스 (Special RSA modulus) : RSA 모듈러스 n=pq는 p = 2p′ +1과 q = 2q′ + 1가 안전한 소수일 때 특별(special)하다고 한다.

CL 서명 스킴은 단일 메시지 서명과 메시지 블록을 위한 메시지 서명 방식을 제안하고 있다. 본 논문에서는 단일 메시지 서명에 관한 흐름을

살펴보도록 하겠다. 단일 메시지 서명의 과정은 키 생성, 서명, 검증의 단계를 거쳐서 인증된다.

- 메시지 공간 (Message space) : l_m는 보안 파라미터이다. 메시지 공간은 길이 l_m인 이진 스트링으로 구성되고, 그 범위는 $[0, 2^{l_m}]$에서의 정수 집합이다.
- 키 생성 : 1^k를 입력 받아 길이 $l_n = 2k$인 특별한 RSA 모듈러스 n=pq와 p = 2p′ +1, q = 2q′ + 1를 선택한다. 또한 균등하고 랜덤하게 $a,b,c \in QR_n$을 선택한다. 키 생성을 통해 공개키 $PK = (n, a, b, c)$ 비밀키 $SK = (p)$을 출력한다.
- 서명 알고리즘 : m을 입력 받아 길이가 $l_e = l_m + 2$인 랜덤 소수 e와 길이가 $l_s = l_n + l_m + l$의 소수 s를 선택한다. 여기서 l은 보안 파라미터 이다. 위의 값을 이용하여 $v^e = a^m b^s c \bmod n$를 계산 할 수 있다.
- 검증 알고리즘 : $Tuple(e, s, v)$는 메시지 공간 내에서 메시지 m에 대한 서명이라는 것을 검증하기 위해서 $v^e = a^m b^s c \bmod n$의 값을 확인하고, $2^{l_e - 1} < e < 2^{l_e + 1}$이라는 것을 확인한다.

CL 서명 스킴은 Cramer-Shoup[16] 스킴과 비교 하였을 때, 효율성 면에서는 떨어지지만 스킴의 기법 및 활용성이 우수하다는 평가를 받고 있다.

3. 모바일 환경을 위한 개선된 DAA 프로토콜

TCG에서 제안된 DAA 프로토콜은 신뢰 컴퓨팅(Trusted computing) 환경의 원격 인증(Remote attestation) 기능에 사용될 것이다. 하지만 현재의 DAA는 많은 계산량을 요구하고 있어 사용 및 구현이 힘들다는 단점을 가지고 있다.

현재 PC에 대한 신뢰 컴퓨팅 기술을 연구하고 있는 TCG는 앞으로 모바일 기기 및 환경에 대한 신뢰 컴퓨팅 기술을 확대할 계획[15, 18, 19, 20]을 가지고 있다. 모바일 환경에 맞는 원격 인증 기능을 제공하기 위해서는 DAA 프로토콜에 대한 수정을 통해 경량화 및 최적화 되어야 한다.

따라서 이 장에서는 기존의 DAA 프로토콜에 대해서 분석하고 분석결과를 통해서 DAA의 단점을 제시한다. 또한 Camenisch와 Lysyanskaya의 서명 스킴과 영지식 스킴을 기반을 기반으로 하여 DAA의 단점을 보완하고 프로토콜을 경량화 시킨 수정된 DAA 프로토콜을 제안한다.

3.1 DAA (Direct Anonymous Attestation)

DAA[6]는 영지식 증명 (Zero knowledge Proof) 과 그룹 서명의 스킴을 사용하여 익명성을 제공함으로써 익명성을 보장하고, 사용자의 인증을 할 수 있는 서명 스킴을 제공하기 위해서 제안되었다.

DAA (Direct Anonymous Attestation) 는 TCG (Trusted Computing Group) 에서 디자인된 그룹 서명 스킴의 하나로 플랫폼을 사용하는 사용자에 대한 프라이버시를 보호하면서 동시에 TPM의 하드웨어의 인증을 제공하는 서명기법이다. TCG에서 제안한 DAA는 다음과 같은 특징을 가

지고 있다.

- 직접 증명 (Direct Proof) : TTP (Trusted Third Party) 를 따로 두지 않고도 영지식 증명을 이용하여 TPM과 검증자의 통신상에서 TPM의 인증이 가능하다.
- 익명성 지원(Anonymous): 어떠한 서명에 대해서 서명자의 신원이 밝혀 지지 않는다.
- 그룹서명의 기능을 사용하지만 서명을 개봉(Open)할 수 있는 능력은 없다.
- Rogue TPM을 찾는 방법이 존재한다.
- Camenisch-Lysyanskaya 서명 기술을 기반으로 한다.
- Strong RSA Assumption 및 Decisional Diffie-Hellman Assumption 을 사용하므로 랜덤 오라클 모델에서 안전하다.
- 영지식 증명을 기반으로 하고 있다.
- 인증서 소유에 대한 증명에 이산 대수를 적용한다.
- 증명(Proof)을 서명으로 전환하게 하기 위해서 Fiat-Shamir 가 제안한 방식을 사용한다.

3.1.1 DAA에 사용된 기술

DAA는 그룹서명의 이점과 Camenisch-Lysyanskaya 서명 기술의 이점을 적용하였다. 이에 따라 DAA는 Strong RSA 가정과 DDH(Decisional Diffie-Hellman) 가정을 기반으로 랜덤 오라클 모델에서 안전한 새로운 서명 방식으로 제안되었다. DAA의 이해를 위해서 DAA에 이용된 기술에 대해서 설명을 하고자 한다. DAA에 적용된 기술을 다음과 같다.

- 영지식 증명 : 증명자와 검증자 사이에서 증명자가 가진 비밀 정보를 검증자에게 알리지 않고 어떠한 대화나 프로토콜을 통하여 비밀정보에 대한 소유를 증명하는 것에 대해 정보를 교환하는 것을 말한다. 영지식 증명의 기본 아이디어는 아래 그림 4와 같다.

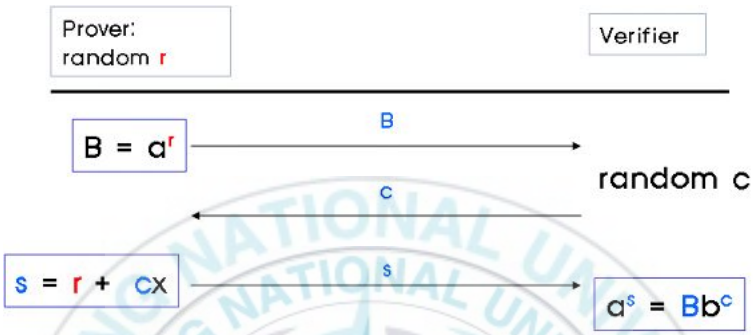

그림 4. 영지식 증명 방식

- 랜덤 오라클 모델 : 랜덤 오라클 모델은 1986년 비공식적으로 Fiat와 Shamir에 의해서 처음 소개되었고 Bellare와 Rogaway에 의해 더욱 정규화 되어 사용되고 있다. 그 이후 수많은 암호학 스킴의 안전성의 증명에 사용되고 있다. 랜덤 오라클 모델은 주로 암호학에서 어떤 암호학적 스킴의 안전성 분석에 사용되고 있다. 스킴의 증명에서 실제로 구현이 가능한 수학적 함수를 제시하지 못 할 때 주로 사용된다. 일례로 높은 랜덤의 특징을 가지는 해시 함수의 출력을 가지는 스킴 나타낼 때 사용된다. 이를 통해 공격자가 스킴의 수학적으로 강력한 알고리즘(이산대수, 인수분해)을 해결하는 것이 불가능하다는 것을 나타낸다.

- Camenisch-Lysyanskaya 서명 스킴 : 암호화 애플리케이션에서의 중

심적인 기술로써, 중요문서를 인증하기 위한 서명을 생성하는 스킴이다. Strong RSA assumption이라는 수학적 기법에 기반을 두고 있고 강한 암호학적 강도를 가지고 있다. 주로 특정 메시지나 값에 대한 서명을 생성하거나 연쇄적으로 다른 애플리케이션에서 사용되기 위한 서명 블록을 만들기 위해 사용된다.

3.1.2 DAA의 구성

DAA는 정해진 디바이스(TPM)들 끼리 미리 약속된 프로토콜을 통해 정보를 주고받음으로써 사용자의 익명성을 지키면서 인증을 할 수 있게 해준다. DAA 프로토콜에 참여하는 주체들은 다음과 같다.

- TPM / Host: 키와 패스워드, 그리고 디지털 인증서를 저장하는 하드웨어 모듈이다.
- 발행자 (Issuer): 인증서를 발급해주는 역할을 하는 주체이다. TPM과의 통신 프로토콜을 통하여 현 TPM의 무결성이 보장되면 플랫폼에 인증서(credential)를 교부해주는 역할을 한다.
- 검증자 (Verifier): TPM / Host는 발급받은 인증서를 사용하여 서명을 생성하고 그 서명은 검증자에게 유효성을 검증(verify) 받는다.

DAA는 그룹 서명을 기반으로 한 서명 방식을 제안하고 있다. 그룹 서명의 절차는 앞에서 설명하였듯이 시작(Setup), 준비(Join), 서명(Sign), 검증(Verify), 익명성 파기(Open)로 나눌 수 있다. 하지만 DAA는 사용자의 익명성을 제공하기 위해 그룹 서명의 익명성 파기(Open) 부분을 제외시켰다. 다음은 DAA에서 이용하는 그룹 서명의 절차 방식에 대한 설명이

다.

- 시작(Setup) : Fiat-Shamir 휴리스틱을 사용하여 발행자의 공개키와 비밀키를 작성한다.
- 준비(Join) : TPM은 DAA 발행자에게 N_1^f의 형태의 익명(pseudonym)을 전달하여 자신이 비밀 정보 f를 가지고 있다고 증명하고, 발행자를 통해 v''의 값을 받아서 인증서(credential) 을 발급받는다.
- 서명(Sign) : TPM은 발급받은 인증서(credential)를 이용하여 메시지 m을 서명한다.
- 검증(Verify) : DAA 검증자에서 서명 과정에서 생성한 서명의 유효성을 검증 받는다.
- 익명성 파기(Open) : 프라이버시 보호를 위해서 그룹 서명 스킴과 달리 DAA는 익명성 파기(Open) 연산을 제공하지 않는다.

3.1.3 DAA와 Privacy CA의 비교

TCG는 TPM 스펙을 정의하면서 1.1 버전에는 원격 인증(remote attestation)을 제공하기 위해서 Privacy CA를 제안하였고, 1.2 버전에서는 Privacy CA의 단점을 보완하기 위해서 DAA를 제안하였다. 본 절에서는 DAA와 Privacy CA의 비교를 통해서 DAA의 장점을 알아보고자 한다.

(1) Privacy CA

TPM 스펙 1.1에서 정의된 Privacy CA는 다음과 같은 특징을 지닌다.

- Trusted Third Party (Privacy CA)를 필요로 한다.
- 각 TPM은 고유의 EK(Endorsement key)를 보유하고 있다
- Privacy CA도 유효한 TPM 식별을 위해서 해당 TPM들의 EK에 대한 리스트를 보유하고 있다.

Privacy CA를 이용한 원격 인증 방식은 그림 5의 순서대로 이루어진다. 원격 인증 방식의 자세한 순서는 다음과 같다.

- TPM은 EK(Endorsement Key)로 서명된 AIK (Attestation Identity Key)를 Privacy CA로 보낸다.
- Privacy CA는 AIK의 유효성을 체크하고, 유효하다면 인증서를 발행한다.
- 인증서를 받은 TPM은 이를 이용하여 자신을 인증시킬 수 있다.

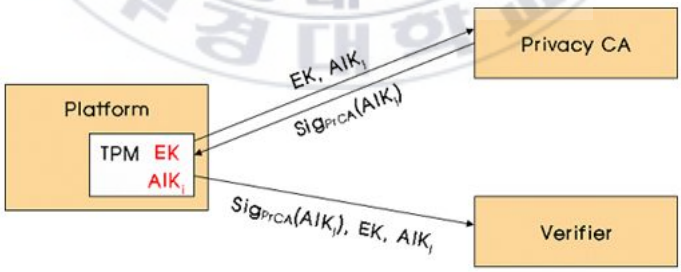

그림 5. Privacy CA를 이용한 인증 순서

Privacy CA의 이용은 기존의 PKI의 개념을 가져와 구현에 대한 효율성이 있다. 하지만 모든 트랜잭션에 Privacy CA가 참여해야 하므로, CA의

부하가 크고, 기존의 보안 방식보다 안전하다는 보장이 없다. 또한 Privacy CA와 검증자가 결탁하면 검증자는 CA의 자료를 불건전하게 이용할 수 있는 단점을 가지게 된다. 이러한 문제점을 해결하기 위해서 TCG은 TPM 스펙 1.2 버전에서는 Privacy CA를 제외시키고 DAA를 적용하였다.

(2) DAA

DAA는 Privacy CA의 단점을 보완하기 위해서 그룹서명, Identity escrow, 인증 시스템(credential system)을 사용하였다. 이들의 적용을 통해서 DAA가 가지는 특징은 다음과 같다.

- 시스템 사용자의 프라이버시를 지켜주면서 플랫폼에 대한 원격 인증을 수행할 수 있다.
- Privacy CA와 달리 추가적인 TTP의 필요성을 제거했다.
- 불법적인 TPM을 감지하기 위한 Rogue Tagging 기능을 갖추었다.

Privacy CA를 DAA로 바꾸면 그림 6과 같은 방식으로 변경되어 진다. DAA의 인증 시나리오는 다음과 같다.

- 호스트는 발행자에게 그룹에 대한 멤버십을 요청한다.
- 발행자는 인증서 (Attestation identity credential) 를 교부한다.
- 인증서를 받은 Host는 그것을 사용하여 익명성을 가지고 검증자에게 자신을 인증시킬 수 있다.

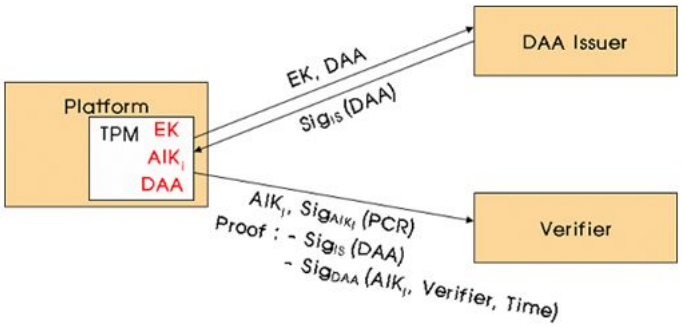

그림 6. DAA를 이용한 인증 순서

3.1.4 DAA 프로토콜 분석

DAA의 프로토콜은 시작(Setup), 준비(Join), 서명(Sign), 검증(Verify) 이렇게 4단계로 구분된다. 본 절에서는 DAA의 흐름을 파악할 수 있도록 DAA의 4가지 과정에 대한 설명을 하고자한다.

(1) 시작(Setup) 과정

시작(Setup) 과정은 발행자(Issuer) 단독으로 실행하는 과정으로 결과물로는 발행자(Issuer)의 공개키와 개인키가 있다. 시작(Setup)은 다음과 같이 이루어지며, 그림 7은 시작(Setup) 과정을 간략화한 모습을 나타낸다.

(가) 발행자는 $p = 2p' + 1$, $q = 2q' + 1$을 만족하는 RSA 모듈러스 n을 선택한다.

(나) 이차 잉여 모듈러 n 그룹의 QR_n에 대한 랜덤 생성자 g′를 선택한다.

(다) 랜덤 정수 6개를 선택하여 랜덤 생성자를 이용하여 g, h, S, Z, R_0, R_1을 계산한다.

(라) 발행자는 g, h, S, Z, R_0, R_1의 값이 제대로 계산되었는지를 분석한다.

(마) 발행자는 정의된 길이를 만족하는 소수(Γ, ρ, γ)를 선택한다.

(바) 발행자는 마지막으로 공개키 $(n, g', g, h, S, Z, R_0, R_1, \Gamma, \rho, \gamma)$를 공개한다.

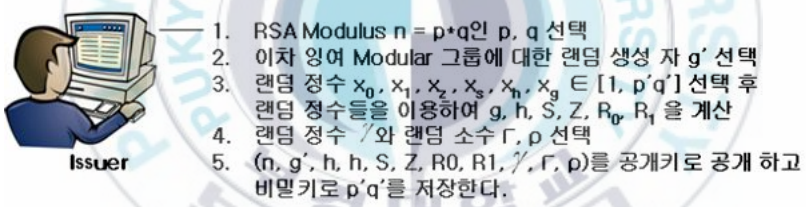

그림 7. 발행자의 Setup 과정

발행자의 공개키가 공개되면, 준비(Join), 서명(Sign)과정에서 발행자의 공개키를 이용하게 되는데 발행자의 공개키의 유효성을 검증하기 위해서는 다음과 같은 과정을 따른다. 그림 8은 공개키 유효성 검사에 대한 과정을 나타낸다.

(가) g, h $\in$ < g'>, S, Z $\in$ < h >, R_0, R_1 $\in$ < S >에 대해서 검증한다. g, h, S, Z, R_0, R_1이 제대로 생성이 되지 않았다는

것은 TPM/Host의 보안이 안전하지 않다는 것을 의미한다.

(나) (Γ, ρ, γ)가 소수인지를 확인한다.

(다) 공개키 파라미터의 길이를 검증한다.

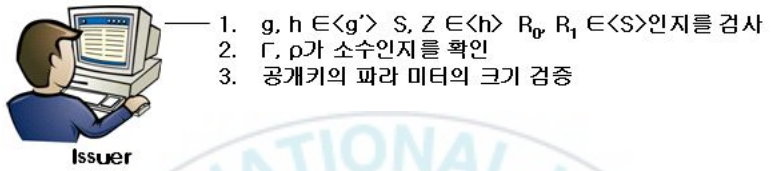

그림 8. 공개키 검증 과정

(2) 준비(Join) 과정

준비(Join)과정은 TPM을 소유한 사용자가 DAA를 이용하기 위해서 TPM의 인증서를 획득하는 과정이다. 이 과정을 통해서 TPM은 자신의 비밀 정보 f, v'를 생성 하고, 발행자로부터 받는 v'' (v = v' + v'')를 통해 TPM은 자신의 내부에 (f_0, f_1, v)를 저장한다. 많은 계산 량과 TPM 기기의 한계로 인해 Join 과정은 TPM과 TPM을 지닌 Host에게 계산 량을 나누어서 처리한다. 준비(Join) 과정은 다음은 준비(Join)에 대한 과정을 나타내고 있다.

(가) Host는 발행자의 베이스네임을 이용하여 $\zeta_I = (H(1\|bsn_I))^{(\Gamma-1/\rho)} \bmod \Gamma$을 계산하고, TPM은 ζ_I의 유효성을 판단한다.

(나) TPM은 비밀정보 f를 생성하여 f_0($f_0 = LSB_{l_f}(f)$)와 f_1 ($f_1 = CAR_{l_f}(f)$)으로 구분하고 랜덤 수 v′를 생성한다. 비밀정보 f_0, f_1, v′를 다음 식을 이용하여 U와 N_I를 계산한 후 발행자에게 전송한다.

$$U := R_0^{f_0} R_1^{f_1} S^{v'} \bmod n, \quad N_I := \zeta_I^{f_0 + f_1 2^{l_f}}$$

(다) 발행자는 rogue list(불법 TPM에 대한 정보가 들어 있는 리스트)를 이용하여 U, N_I를 전송한 TPM이 불법적인지 아닌지를 판단하고 불법적이라면 프로토콜을 취소한다.

(마) TPM과 발행자는 다음과 같은 지식 서명(Signature Proof of Knowledge : SPK)을 수행한다. 지식 서명은 Camenisch가 정의한 것으로 영지식 증명을 통해서 TPM이 비밀정보를 가졌다는 것을 비밀 정보를 발행자에게 전송하지 않고 증명하는 것을 말한다. 영지식 증명은 (나)에서 생성한 U와 N_I를 증명하기 위해서 새로운 $\tilde{U}, \tilde{N}_I$ 을 만들고, s = r + cx ($e.g.\ s_{f_0} = r_{f_0} + cf_0$)의 기반의 비밀 값들을 만들어 계산한다.

$$SPK \left\{ \begin{array}{l} (f_0, f_1, v') : U \equiv \pm R_0^{f_0} R_1^{f_1} S^{v'} \pmod n \ \wedge \\ N_I := \zeta_I^{f_0 + f_1 2^{l_f}} \pmod \Gamma \ \wedge \\ f_0, f_1 \in \{0,1\}^{\ell_f + \ell_\Phi + \ell_H + 2} \ \wedge \ v' \in \{0,1\}^{\ell_f + \ell_\Phi + \ell_H + 2} \end{array} \right\}$$

(바) TPM과 발행자 사이의 SPK{(f_0, f_1, v′)}의 증명이 완료되면 발행자는 TPM이 비밀정보를 알고 있다는 것을 믿게 된다. 인증서(credential)를 위해서 발행자 아래의 A를 계산하고 Host

와 지식서명(SPK)을 통해서 인증서 생성에 필요한 v'' 정보를 전송한다. 발행자와 Host 사이의 지식서명(SPK)은 다음과 같다.

$$A := (\frac{Z}{US^{v''}})^{1/e} \bmod n$$

$$SPK\left\{(d) : A \equiv \pm(\frac{Z}{US^{v''}})^{d} (\bmod n)\right\}(n_h)$$

(사) 발행자와 Host 사이에 영지식 증명이 끝이 나면 발행자는 Host에게 (A, e, v'')를 전송하고, Host는 TPM에게 v'' 정보를 전송한다.

(아) TPM은 전송받은 v''를 이용하여 $v = v' + v''$ 를 계산하고 비밀정보 (f0, f1, v)를 저장한다.

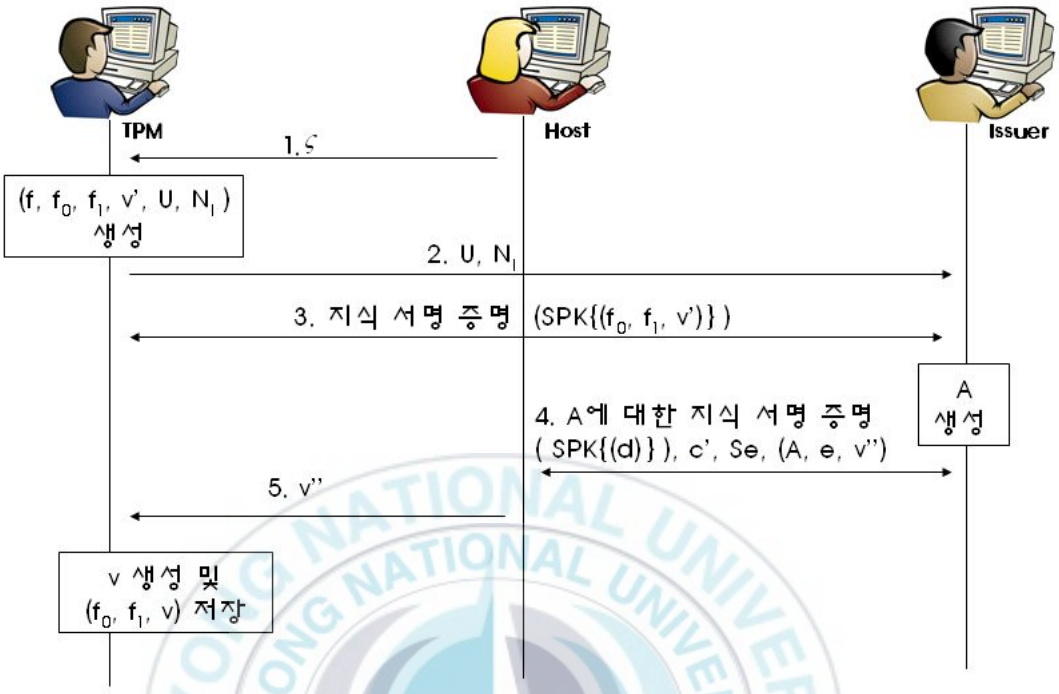

그림 9. 준비(Join) 과정

(3) 서명(Sign) 과정

서명(Sign)과정은 TPM/Host를 소유한 사용자가 어떠한 메시지를 서명하기 원할 때 (2)의 준비(Join) 프로토콜에서 얻은 (f_0, f_1, v)와 발행자의 공개키 그리고 서명될 메시지를 이용하여 전자서명을 생성하는 과정이다. 서명(Sign)의 과정은 다음과 같다.

(가) Host는 검증자의 베이스 네임을 통해 ζ을 계산하고, TPM은 이 ζ의 유효성을 검증한다. ζ는 검증자의 베이스 네임 구성에 의해서 다르게 구해질 수 있다.

(나) Host는 랜덤 정수 w, r를 선택하여 T_1, T_2를 계산한다. TPM은 Host로부터 전송받은 ζ을 이용하여 N_v를 계산 한 후 이를 Host에 전송한다.

$$T_1 := Ah^w \bmod n, \quad T_2 := g^w h^e (g')^r \bmod n$$
$$N_V := \zeta^{f_0 + f_1 2^{l_f}} \bmod \Gamma$$

(다) TPM과 Host는 전자서명을 생성하기 위해서 다음과 같은 지식 서명(SPK)을 생성한다.

$$SPK\left\{\begin{array}{l}(f_0, f_1, v, e, w, r, ew, ee, er): \\ Z \equiv \pm\, T_1^e R_0^{f_0} R_1^{f_1} S^v h^{-ew} \pmod n \ \wedge \\ T_2 \equiv \pm\, g^w h^e g'^r \pmod n \ \wedge \\ N_v \equiv \zeta^{f_0 + f_1 2^{l_f}} \pmod n \wedge f_0, f_1 \in \{0,1\}^{\ell_f + \ell_\Phi + \ell_H + 2}\end{array}\right\}(n_t \| n_v \| b \| m)$$

(라) $SPK(f_0, f_1, v, e, w, r, ew, ee, er)$을 진행하면 TPM은 영지식에 필요한 $s_{f_0}, s_{f_1}, s_v, s_e, s_{ee}, s_w, s_{ew}, s_r, s_{er}$을 생성하게 되고, Host는 이 값들을 이용하여 $\widetilde{T_1}, \widetilde{T_2}$에 대한 영지식 증명을 완료한다.

(마) 영지식 증명이 완료되면 Host는 영지식 증명에 사용되었던 값들을 이용하여 서명 σ을 생성한다.

$$c_h := H((n \| g \| g' \| h \| R_0 \| R_1 \| S \| Z \| \gamma \| \Gamma \| \rho) \| \zeta \| (T_1 \| T_2) \| N_v \| (\widetilde{T_1} \| \widetilde{T_2} \| \widetilde{T'_2}) \| \widetilde{N_v} \| n_v) \in [0, 2^{\ell_H} - 1]$$
$$c := H(H(c_h \| n_t) \| b \| m) \in [0, 2^{\ell_H} - 1]$$

$$\sigma := (\zeta, (T_1, T_2), N_v, c, n_t, (s_v, s_{f_0}, s_{f_1}, s_e, s_{ee}, s_w, s_{ew}, s_r, s_{er}))$$

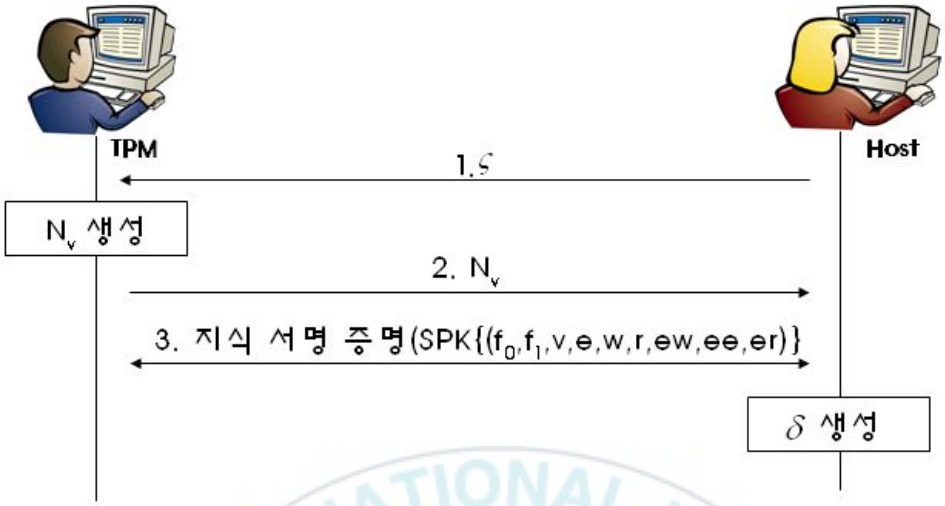

그림 10. 서명(Sign) 과정

(4) 검증(Verify) 과정

검증(Verify) 과정은 서명(Sign)과정에서 생성된 서명과 검증(Verify)과정에서 생성되는 값이 같은지를 비교하여 서명에 대한 유효성을 판단한다. 검증(Verify)과정은 다음과 같이 이루어진다.

(가) 검증자는 검증을 원하는 서명의 정보를 이용하여 $\hat{T}_1, \hat{T}_2, \hat{T'}_2, \hat{N}_v$를 계산한다. $\hat{T}_1, \hat{T}_2, \hat{T'}_2, \hat{N}_v$를 이용하여 c'를 계산하고, 계산이 완료되면 서명 σ의 c값과 비교하여 같다면 다음 과정을 수행하고 아니라면 검증 요청을 무시한다.

$$\widehat{T_1} := Z^{-c} T_1^{s_e + c2^{\ell_e - 1}} R_0^{s_{f_0}} R_1^{s_{f_1}} R^{s_v} h^{-s_{ew}} \bmod n$$
$$\widehat{T_2} := T_e^{-c} g^{s_w} h^{s_e + c2^{\ell_e - 1}} g'^{s_r} \bmod n$$
$$\widehat{T_2} := T_e^{-s_e + c2^{\ell_e - 1}} g^{s_{ew}} h^{s_{ee}} g'^{s_{er}} \bmod n$$
$$\widehat{N_v} := N_v^{-c} \zeta^{s_{f_0} + s - f_1 2^{\ell_f}} \bmod \Gamma$$

$$c' := H((n \| g \| g' \| h \| R_0 \| R_1 \| S \| Z \| \gamma \| \Gamma \| \rho) \| \zeta \| (T_1 \| T_2) \| N_v \| (\widehat{T_1} \| \widehat{T_2} \| \widehat{T'_2}) \| \widehat{N_v} \| n_v) \in [0, 2^{\ell_H} - 1]$$

(나) 검증자는 베이스 네임을 이용하여 ζ를 구한다.

(다) 검증자는 Rogue list에 존재하는 모든 (f_0, f_1)에 대해 TPM이 전송한 서명에 존재하는 Nv와 같은지를 계산한다. 만약 같은 값이 있다면 서명이 유효하지 않다고 판단하고. 같은 값이 없다면 서명이 유효하다고 판단한다.

1. $\hat{T_1}, \hat{T_2}, \hat{T_3}, \hat{N_v}$ 을 계산
2. $\hat{T_1}, \hat{T_2}, \hat{T_3}, \hat{N_v}$ 을 이용하여 C 계산하여 (4)에서 만들어진 서명(δ)의 C와 같은지 비교
3. 같다면, 베이스 네임에 대한 ς 를 구함
4. Rogue 리스트의 모든 (f_0, f_1)에 대해 N_v를 비교하여 NV와 같은 값이 있다면 유효하지 않다고 답함

그림 11. 검증(Verify) 과정

3.1.5 DAA 프로토콜의 취약점

본 절에서는 DAA가 가지는 취약점에 대해서 논의하고자 한다. DAA는

영지식 증명 등을 이용하여 다른 그룹 서명 기법보다 많은 연산을 요구하고 있다. 또한 베이스 네임의 결정 방식을 2가지로 두어 베이스 네임에 따른 불법 TPM의 위장 문제를 발생 시킬 수 있다. 마지막으로 TPM을 검증하는 검증자가 발행자의 결탁에 따른 프라이버시 문제점을 생각할 수 있다. 본 논문에서는 이러한 문제점에 대해서 살펴보고, 문제점을 해결하는 해결 방안을 제시할 것이다. DAA가 가지는 문제점은 다음과 같다.

(1) 많은 계산 량에 대한 문제점

DAA는 위의 프로토콜 분석에서 보았듯이 영지식 증명을 이용하여 많은 계산 량을 요구한다. 이러한 많은 계산 요구량은 DAA의 상용화를 늦추고 있으며 구현에 큰 어려움을 제공한다. 본 연구실에서 개발하고 있는 자바 기반의 DAA 프로토콜 모듈에서 DAA의 발행자의 서명 과정을 테스트 하면 1시간이 넘게 계산을 하고 있음을 볼 수 있었다. 따라서 PC 또는 모바일 환경에서 DAA를 효율적으로 사용하기 위해서는 계산량을 줄인 수정된 DAA가 필요하다.

(2) 베이스 네임 결정에 대한 문제점

DAA를 제안한 저자들은 불법 TPM 검색에 이용되는 베이스 네임(bsn)의 생성 방법을 두 가지로 제시하였고, 이 방법 중 개발자에 맞는 방법을 사용하도록 제안하고 있다. 베이스 네임 생성 방법의 종류는 다음과 같다.

- 발행자와 검증자에서 베이스네임 생성 : TPM에서 계산하는 ζ의 기초가 되는 베이스네임의 값을 발행자와 검증자가 생성한 이름을 사용하

여 계산하는 방법이다.

- TPM이 생성한 값을 베이스네임으로 이용 : TPM에서 랜덤으로 생성한 값을 베이스 네임으로 사용하는 방법이다.

만약 DAA 프로토콜에서 TPM이 랜덤으로 생성한 값을 베이스 네임으로 사용한다면 불법 TPM도 새로운 베이스 네임을 생성하여 새로운 ζ와 N값을 구할 수 있다. 만약 이것이 가능하다면 불법 TPM이 불법 TPM이 아닌 것처럼 위장할 수 있는 문제점이 발생한다.

(3) 프라이버시에 대한 문제점

DAA 프로토콜은 발행자와 검증자가 사용자(Host/TPM)의 프라이버시를 깨기 위해서 공모를 할 수 있다. 발행자와 검증자가 공모를 통해서 동일한 베이스 네임을 가진 후 프로토콜을 수행하면 인증서를 발행받기 위해서 발행자에게 보내지는 ζ의 결과와 검증 과정에서 요구되는 검증자와 관련된 ζ의 결과가 같게 된다. 이로 인해 검증자는 사용자가 언제 인증서를 발행 받았고, 검증을 요구하는지를 알 수 있게 되므로 프라이버시에 문제가 발생 할 수 있다. 그림 12는 발행자와 검증자의 공모를 통해 생기는 프라이버시 문제를 나타낸다.

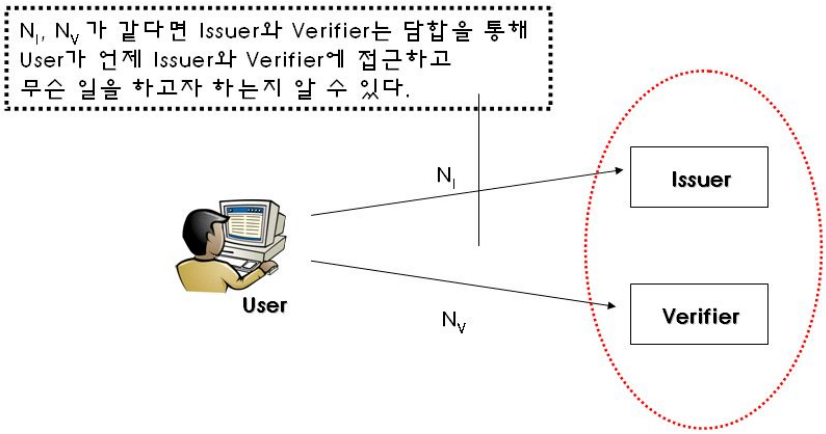

그림 12 발행자와 검증자의 공모를 통한 프라이버시 문제

3.2 불법 TPM 검색 및 프라이버시 문제 해결

DAA를 제안한 저자들은 불법 TPM을 찾는데 사용되는 베이스 네임의 선택을 두 가지 방식으로 제안하여 TPM이 랜덤으로 생성한 값을 베이스 네임으로 사용한다면 불법 TPM도 새로운 베이스 네임을 생성할 수 있는 문제점을 지니고 있다. 이는 불법 TPM이 불법이 아닌 것처럼 위장하여 서비스를 제공받을 수 있음을 의미한다. 본 논문에서는 이러한 문제점을 해결하기 위해서 DAA 프로토콜에서 사용될 불법 TPM 검색 방식을 변경하였다. 변경된 불법 TPM 검색 방식은 다음과 같다.

$$\zeta_I = H(g \| U \| 0) \pmod{n}, N_I = \zeta_I{}^x \pmod{n}$$

$$\zeta_V = H(g \| U \| 1) \pmod{n}, N_V = \zeta_V{}^x \pmod{n}$$

ζ를 구할 때 사용되는 베이스 네임을 사용자 TPM의 EK 비밀키의 일

부분을 사용하여 만들어진 U값을 사용함에 따라 베이스 네임이 고정되어 있어 불법 TPM이 새롭게 베이스 네임을 만들 수 있는 환경을 없앴다. 그리고 발행자(ζ_I부분)와 검증자(ζ_V부분)의 계산 방식을 따로 두어 발행자와 검증자의 결탁으로 인해 사용자의 프라이버시 문제가 발생하는 것을 막았다. 위의 식은 DAA의 저자들이 제안한 랜덤 값을 이용하여 베이스 네임을 이용하지 않고 사용자의 비밀정보의 일부를 이용하여 ζ를 구하기에 새로운 프라이버시 문제점이 발생할 것처럼 보이지만 U의 값은 2048 비트의 RSA 모듈러스 n을 통해서 모듈러화 되기 때문에 사용자의 비밀정보 x를 모른다면 U의 값을 구하기 어렵기 때문에 공격자는 U 값이 누구의 것인지 알 수 없게 된다. 따라서 새롭게 제시한 방법에 대한 공격자의 프라이버시 공격은 상당히 비효율적일 것으로 예상된다.

3.3 성능 향상을 위해 개선된 DAA 프로토콜

현재의 DAA 프로토콜은 모바일이 가지는 환경적 제약을 고려하지 않은 채 연구되어 모바일 환경에 바로 사용하기 어려울 것으로 예상된다. 따라서 본 논문에서는 모바일 환경에 사용할 수 있도록 수정된 DAA 프로토콜을 제시한다. 이 프로토콜에서 필요한 보안 파라미터는 다음과 같다. 보안 파라미터는 Camenisch-Lysyanskaya의 서명 기법 중 단일 메시지 서명 기법에 이용된 파라미터 크기를 적용하였다.

- $\ell_n(1024)$: RSA 모듈러스 n의 길이
- $\ell_x(160)$: TPM이 가지는 EK(Endorsement key)의 비밀키의 부분 정보

길이

- $\ell_H(160)$: 해시 함수(SHA-1) 결과의 길이
- $\ell_c(162)$: 준비(Join)과정에서 필요한 영지식 증명 랜덤 소수 c의 크기
- $\ell_r(1264), \ell_w(1185)$: 서명 증명을 위해 사용되는 랜덤 소수의 크기
- $\ell_v(1185)$: 준비(Join)과정을 통해서 얻어진 비밀 정보 v의 크기

3.3.1 발행자의 시작(Setup) 프로토콜

이 섹션에서는 발행자의 공개키와 비밀키를 만드는 과정을 보일 것이다. 발행자의 공개키는 준비 (Join), 서명 (Sign) 프로토콜의 인자로 사용되며 비밀키는 발행자 (Issuer) 에 의해 보호된다.

프로토콜 1. 수정된 시작 프로토콜

(1) 발행자는 p = 2p′ + 1과 q = 2q′ +1을 만족하는 RSA Modulus n = pq 을 선택한다. 여기에서 p, q, p′, q′는 소수이다.

(2) 발행자는 또한 QR_n 의 랜덤 생성자 g'를 선택한다.

(3) 랜덤 정수 $x_g, x_s \in [1, p'q']$을 만족하는 g, S 값을 계산한다.

$$g = g'^{x_g} \bmod n \quad S = g^{x_s} \bmod n$$

(4) 생성된 g가 <g'>에 S가 <g>에 포함되는지 판단한다.

(5) 발행자는 자신의 공개키 (n, g′, g, S)를 공개키로 공개하고, p′ * q′를 비밀키로 저장한다.

3.3.2. 준비(Join) 프로토콜

준비(Join) 프로토콜은 TPM을 가지고 있는 디바이스(host)의 사용자가 자신의 TPM을 통하여 DAA 인증을 받고자 할 때 사용되는 인증서(credential)를 얻기 위한 과정이다.

본 논문에서는 Camenisch-Lysyanskaya [1, 12]의 서명 기법과 Camenisch-Stadler [1]가 제안한 인증서 생성 기법과 영지식 증명을 응용하여 기존의 준비(Join) 프로토콜을 개선하였다.

프로토콜 2. 수정된 준비 프로토콜

(1) TPM은 영지식 증명에 사용될 Z_n^* 에 속하는 랜덤 정수 w를 선택한다.

(2) TPM은 아래의 식을 이용하여 (x, U, B, N_I)를 계산한다. x는 자신이 소유한 EK(Endorsement key) 비밀키의 부분 정보를 나타낸다. TPM은 랜덤 수 r을 택하여 B의 값을 계산한다. 계산이 완료되면 TPM은 U, B, N_I를 발행자에게 전송한다.

$$x = LSB_{\ell_x}(EK_{sk})$$
$$U = g^x \pmod n,\ B = g^r \pmod n$$
$$\zeta_I = H(g \| U \| 0) \pmod n,\ N_I = \zeta_I{}^x \pmod n$$

(3) 발행자는 rogue list(불법 TPM에 대한 정보가 들어 있는 리스트)를 이용하여 U, B, N_I를 전송한 TPM이 불법적인지 아닌지를 판

단하고 불법적이면 요구를 무시한다.

(4) 발행자는 다음과 같은 서명 지식 증명(Signature Proof of Knowledge : SPK)를 수행한다. SPK의 수행 과정은 아래와 같다.

$$\left\{SPK(x):\ U = \pm g^{x} \pmod{n} \wedge\ N_I = \zeta_I^{x} \pmod{n}\right\}$$

- 발행자는 랜덤 값 c를 선택하여 host에게 전송한다.
- host는 TPM에게 c를 전송하고, TPM은 $s = r + cx$를 선택한다. TPM은 $result = H(a^s \| c)$을 계산하고 host를 통해 s, result를 전송한다.
- 발행자는 BU^s를 계산하고, $result = H(BU^s \| c)$인지를 검사한다. 결과가 일치하면 인증서(credential)를 요구하는 TPM은 합법적이므로 U를 이용하여 TPM의 인증서의 정보 v를 생성한다.

(5) 발행자는 다음 식을 이용하여 TPM의 인증서 정보 v를 생성한 후 Host에 전송한다.

$$v \equiv (U+1)^{1/e} \pmod{n}$$

(6) Host를 통해 v의 정보를 받은 TPM은 (x, v)를 인증서로 생성하여 저장한다.

3.3.3 서명(Sign) 프로토콜

서명 (Sign) 과정은 TPM/host를 소유한 사용자가 어떠한 메시지를 서

명하기를 원할 때 준비(Join) 과정에서 얻은 인증서 정보 (x, v)와 발행자의 공개키를 이용하여 전자서명을 생성하는 과정이다. 본 논문에서는 기존의 DAA 프로토콜을 제안한 저자의 아이디어를 따르고 있다. 하지만 변경된 준비(Join) 과정으로 인해 서명(Sign) 과정에도 계산량을 줄일 수 있었다. 서명(Sign) 과정은 다음과 같다.

프로토콜 3. 수정된 서명 프로토콜

(1) TPM은 자신의 U값을 이용하여 ζ를 생성한다.

$$\zeta_V = H(g \| U \| 1) \pmod{n}$$

(2) TPM은 랜덤 정수 w, r을 선택하여 다음을 계산한다. 또한 TPM은 ㉠에서 생성한 ζ를 이용하여 N_v를 계산한다.

$$T_1 := rg^x S^v \bmod n \quad T_2 := g^w \bmod n$$
$$N_v := \zeta^x \bmod n$$

(3) TPM은 x, r, w 값을 숨기면서 x, r, w 값을 가지고 있다는 것을 증명하기 위해서 r_x, r_r, r_w의 임의 값을 선택 한 후, $\widetilde{T_1}$, $\widetilde{T_2}$를 계산한다. 계산이 완료 되면 결과를 이용하여 c_h를 계산한다.

$$\widetilde{T_1} := T_1^{r_r} g^{r_x} \bmod n \quad \widetilde{T_2} := g^{r_w} \bmod n \quad Z := T_1^r g^x \bmod n$$

$$c_h := H(T_1 \| T_2 \| \zeta_v \| \widetilde{T_1} \| \widetilde{T_2} \| m)$$

(4) TPM은 검증자의 검증에 사용될 비밀 정수 s_x, s_r, s_w를 계산한 후, 다음과 같은 전자서명 σ을 생성한다.

$$s_x = r_x + c_h x \quad s_r = r_r + c_h r \quad s_w = r_w + c_h w$$
$$\sigma = (c_h, U, T_1, T_2, N_v, Z, s_x, s_w, s_r)$$

3.3.4 검증(Verify) 프로토콜

검증(Verify) 과정은 서명 과정에서 생성된 전자서명의 요소들을 이용하여 검증자는 전자서명의 유효성을 판단한다. 상대방의 비밀정보를 모르는 검증자는 영지식 증명을 통해서 상대방의 전자서명을 검증할 수 있다.

프로토콜 4. 수정된 검증 프로토콜

(1) 검증자는 $Z, \hat{T}_1, \hat{T}_2$를 계산하여 c_h' 값을 계산한 후, c_h'와 c_h의 값을 비교 하여 같은지 판단한다.

$$\hat{T}_1 := Z^{-c_h} T_1^{s_r} g^{s_x} \bmod n \quad \hat{T}_2 := T_2^{-c_h} g^{s_w} \bmod n$$
$$c_h' := H(T_1 \| T_2 \| \zeta_v \| \hat{T}_1 \| \hat{T}_2 \| m)$$

(2) c_h'와 c_h의 값이 같으면 다음의 공식을 이용하여 ζ_V, N_I를 구한 후 , rogue list에 존재하는 지를 파악한다. 만약 rogue list에 N_I가 존재한다면 검증과정을 철회 하고, 존재하지 않는다면 TPM 인증서가 유효하다고 답을 한다.

$$\zeta_V = H(g \| U \| 1) \pmod n, N_V = {\zeta_V}^x \pmod n$$

4. 구현 및 평가

이 장에서는 기존의 DAA 프로토콜 및 수정된 DAA 프로토콜의 구현 결과를 제시하고, 수정된 DAA 프로토콜의 안정성 및 효율성을 평가한다.

4.1 전체 구성도 및 구현

DAA 환경을 구현한 시스템의 전반적인 구성은 다음 그림 13과 같다. 구현된 DAA 프로토콜은 TCP/IP 기반의 Java의 Socket을 이용하여 DAA를 구성하는 TPM과 발행자, 검증자들 사이의 통신을 제공하며, 각각의 개체들을 컴포넌트 화하여 기능을 제공한다.

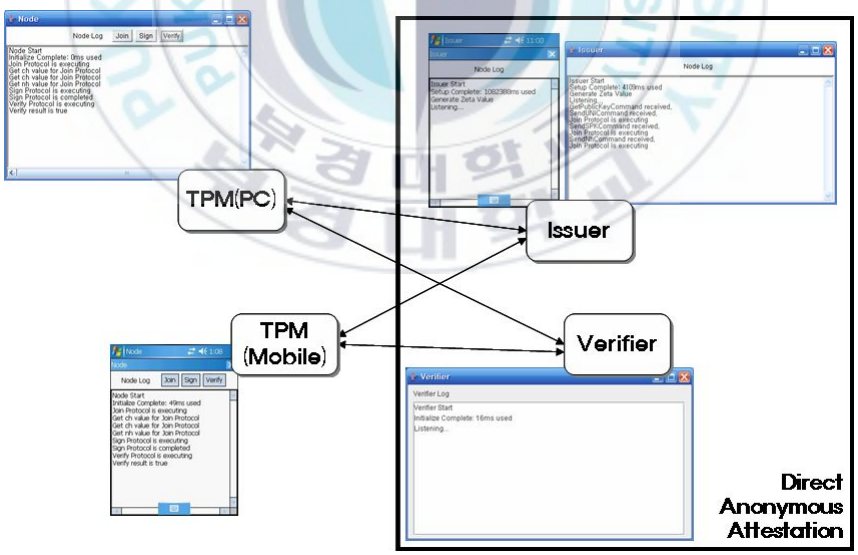

그림 13. DAA 프로토콜 기반의 시스템 구성도

- TPM 컴포넌트는 하드웨어 TPM 기능 중 DAA 기능에 필요한 TPM

기능을 소프트웨어 적으로 구현하여 DAA에 필요한 기능을 수행한다.

- 발행자(Issuer) 컴포넌트는 준비(Join), 서명(Sign), 검증(Verify) 프로토콜에 필요한 발행자의 공개키, 비밀키를 생성하고 영지식 증명에 사용되는 파라미터들을 생성한다.
- 검증자(Verifier) 컴포넌트는 TPM과 발행자 사이에 생성된 인증서와 관련된 전자서명을 검증하는 기능을 수행한다.
- 네트워크(Network) 컴포넌트는 TPM과 발행자, 검증자 사이의 통신을 담당한다. 네트워크 컴포넌트는 서버와 클라이언트 사이의 통신 핸들러(Handler) 구현 및 커맨드(Command) 패턴 적용을 통해서 사용자가 통신 기능을 쉽게 사용할 수 있도록 기능을 제공한다.
- 유틸(Util) 컴포넌트는 키 관리 및 소수 검색 등 발행자와 검증자가 필요로 하는 기능과 관련된 클래스들의 패키지이다.

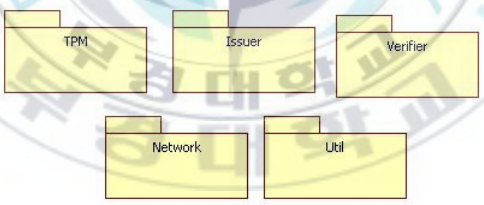

그림 14. 구현된 DAA관련 컴포넌트 구성도

4.1.1 DAA 프로토콜에 대한 요구사항 분석

그림 14를 구성하는 DAA 프로토콜은 그림 15와 같이 DAA 프로토콜이 가져야 하는 요구사항을 만족해야 한다. 요구사항을 나타내는 유즈케이스는 다음과 같다.

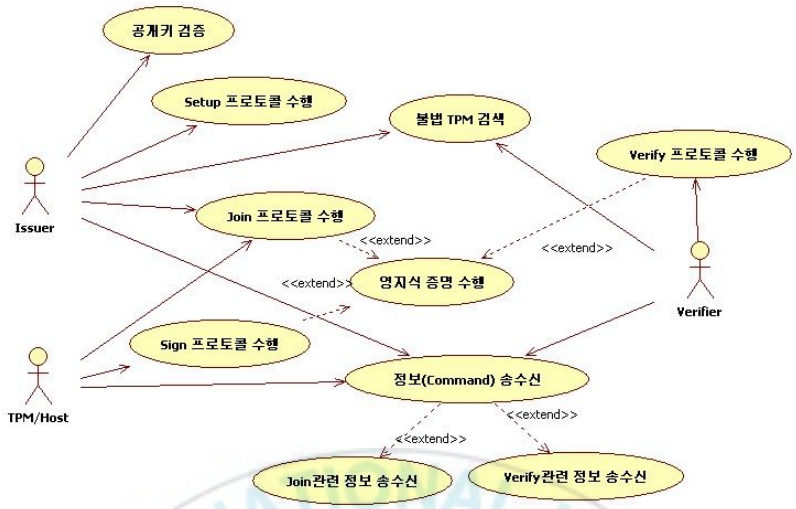

그림 15. DAA 프로토콜이 가지는 요구사항

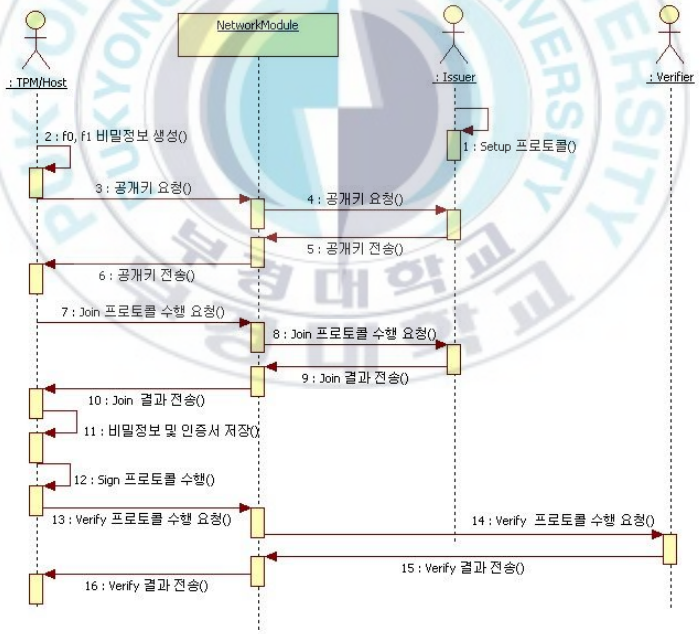

그림 16. DAA 1.0의 순차도

본인이 속한 연구실에서 개발한 DAA 1.0은 DAA에 필요한 기능이 구현되어 DAA를 적용하기 위한 환경에서 테스트할 수 있다.

DAA 1.0은 DAA를 쓴 저자들의 요구사항을 모두 만족하고 있으며 그림 16과 같은 순서를 통해서 DAA 프로토콜이 이루어진다. DAA 프로토콜을 구성하는 TPM, 발행자, 검증자는 모두 Thread를 이용하여 Socket 기반의 통신을 하고, 통신에서는 공통적인 통신기능이 구현된 NetworkModule을 통해서 전자서명 생성 및 검증에 이용되는 각각의 커맨드(Command)를 주고받으면서 DAA 프로토콜을 수행한다.

4.1.2 DAA 프로토콜(DAA 1.0)에 대한 설계 및 구현

DAA 1.0은 DAA 프로토콜을 구성하는 발행자, TPM/Host, 검증자의 구성을 편리하게 하기위해서 인터페이스를 통한 상속을 중심으로 구성되었다. 디바이스는 디바이스의 속성을 저장하고 각 모듈들을 제어하기 위한 Host , DAA프로토콜을 구현한 DAA 모듈, 다른 디바이스와의 통신을 담당하는 Network 모듈로 구성된다.

그림 17은 DAA 프로토콜을 핵심적으로 수행하는 DaaModule과 DaaModule에서 이용하는 클래스들을 나타낸 클래스도이다. DaaModule은 DAA 프로토콜이 가지는 Setup, Join, Sign, Verify에 대한 기능을 객체에 따라서 수행을 한다. 따라서 각각의 발행자, TPM/Host, 검증자는 자신에게 필요한 일을 수행할 때는 DaaModule을 기반으로 수행하게 된다.

또한 그림 18은 각각의 구성원들이 Join, Verify의 기능을 수행할 때 필요한 정보들을 상대방에게 전송할 때 사용되는 Command 관련 클래스도이다. Command들은 Serializable 인터페이스를 구현하여 통신모듈을 통해 객체로 전송된다.

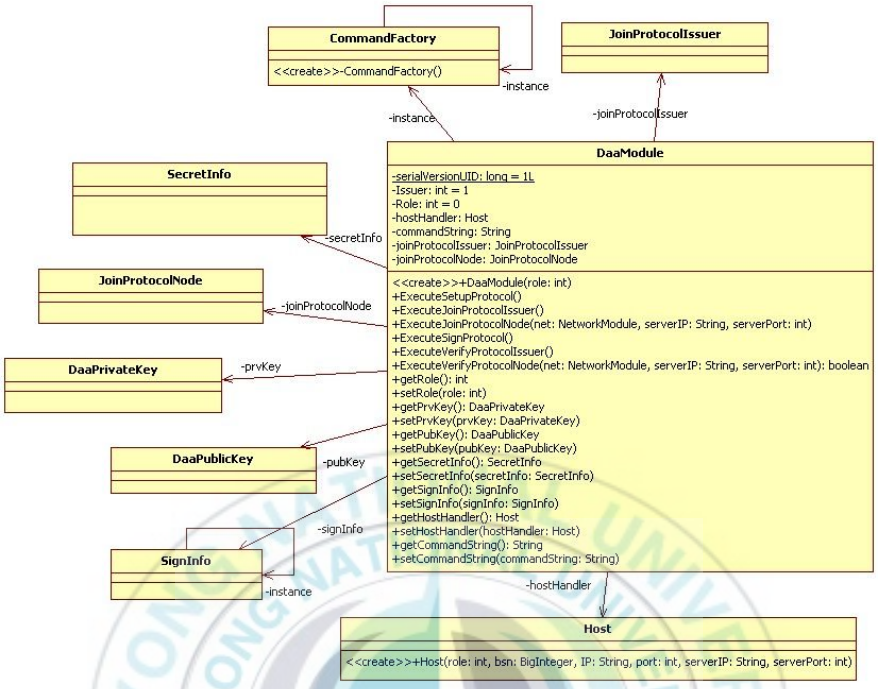

그림 17. DAA 프로토콜의 전반적인 모습을 나타내는 클래스도

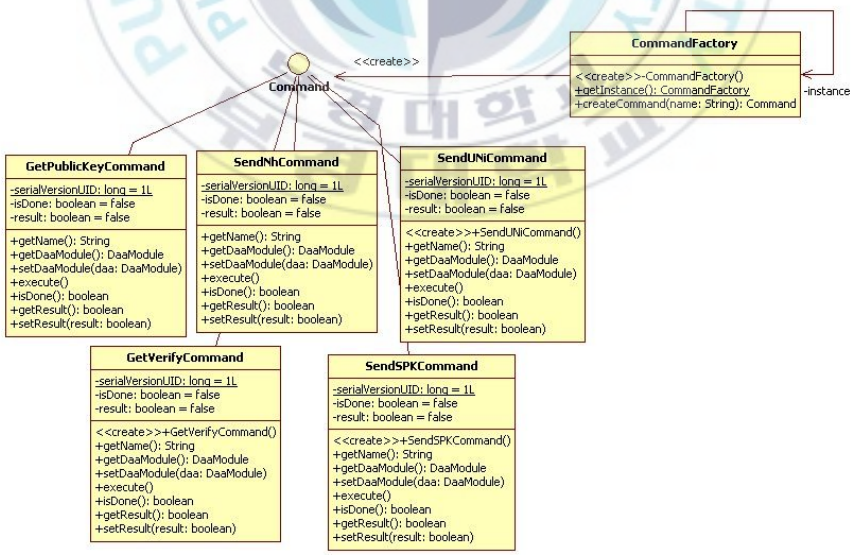

그림 18. Command 관련 클래스도

그림 19는 DaaModule과 Command 클래스 및 프로토콜 클래스들을 이

용하여 나타낸 DAA 프로토콜의 전체 클래스도이다. 각각의 모듈은 가능한 서로의 기능을 공유하지 않고 완전히 분리되어 자신만의 기능을 담당할 수 있도록 설계되었다.

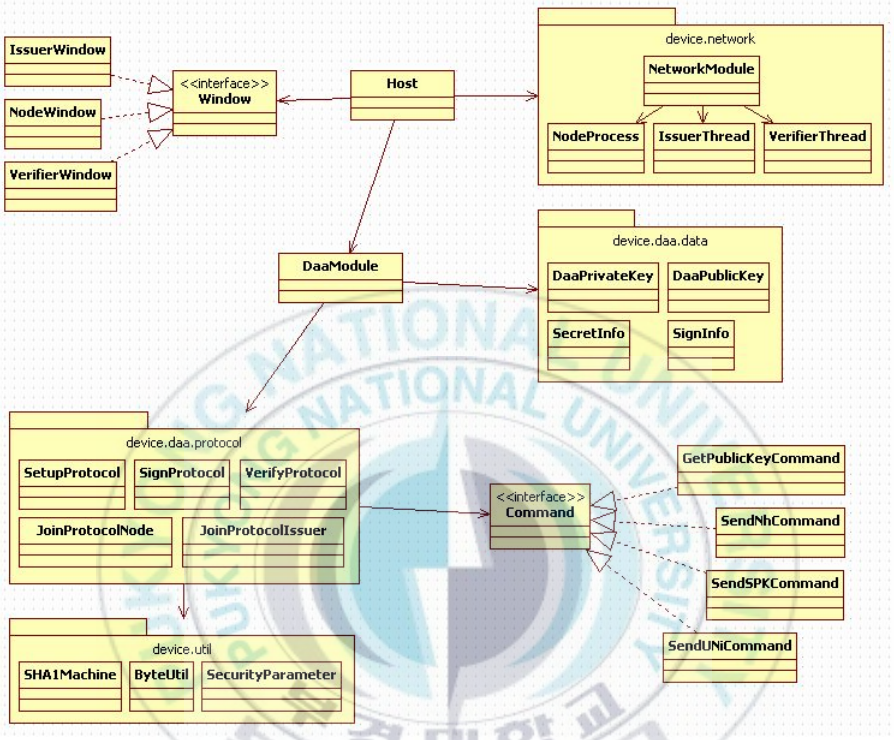

그림 19. DAA 실험 모듈의 전체 클래스도

DAA 관련 컴포넌트들의 개발 환경은 다음과 같다.

(1) TPM

- Language : JDK 1.5.2, XML
- OS : Window XP
- 기타 API : Digest API, JCE/JCA API

(2) TPM Issuer

- Language : JDK 1.5.2
- OS : Window 2003/XP
- 기타 API : JCE/JCA API

(3) DAA Verifier

- Language : JDK 1.5.2
- OS : Window 2003/XP
- 기타 API : JCE/JCA API

4.2 구현 및 제안 기술에 대한 평가

본 절에서는 구현을 통하여 기존의 DAA 프로토콜이 가지는 연산량에 대한 평가와 제안한 수정된 DAA 프로토콜의 안정성 및 효율성을 알아보고, DAA 프로토콜의 영지식 증명이 올바른 수학적 기반을 가지는지 분석한다.

4.2.1 DAA 프로토콜 구현의 평가

본 DAA 1.0은 PC 환경에 최적화되어 구현이 되었다. PC 환경에서 DAA 프로토콜을 돌려봄으로써, 기존의 DAA 프로토콜을 PC 또는 임베디드에서 사용하고자 할 때 어떠한 성능을 발휘하는지 평가하고자 한다. 본 실험에서는 키의 크기가 2048인 경우, 100번씩의 수행을 통해서 수행 성능을 비교하였다. 다음 표1은 프로토콜별 수행 시간을 나타낸다.

표 1. 프로토콜별 수행 시간(2048비트, 단위 : msec)

	Setup	Join	Sign	Verify
최소소요시간	478,844	2,469	2,390	2,234
최대소요시간	522,188	3,062	3,031	2,688
평균소요시간	489,257	2,542	2,409	2,319

표 1의 결과에 따르면, DAA 프로토콜은 대부분의 수행시간이 Setup 과정에서 소모됨을 알 수 있다. Setup 과정은 그룹 서명에 필요한 공개키 생성 및 검증 부분에서 계산양이 많이 요구되며, 이로 인해 긴 시간이 필요하다.

표 2는 실제 사용하게 될 키 및 파라미터의 생성에는 영향을 미치지 않는 검증과정을 포함했을 경우와 제거했을 경우에 대한 Setup 과정의 수행시간을 2048bit와 512bit의 키 크기를 기준으로 측정한 것이다. 컴퓨터 연산을 통해서 만들어진 각각의 보안 파라미터들을 검증하지 않고, 컴퓨터 연산의 능력을 믿으면서 DAA 프로토콜을 PC 환경에서 이용하면 Setup 수행 시간이 급격히 줄어들므로 PC 환경에서는 충분히 기존의 DAA 프로토콜을 이용할 수 있을 것이라는 결론이 내려진다.

표 2. 검증 과정 여부에 따른 Setup 과정의 수행 시간(단위:msec)

	검증과정 포함		검증과정 미포함	
	2048bit	512bit	2048bit	512bit
최소소요시간	478,844	10,656	4,047	987
최대소요시간	522,188	52,141	53,359	32,875
평균소요시간	489,257	19,376	14,207	9,307

그림 20과 21은 검증과정 여부에 따른 Setup 과정의 수행 시간을 그래프로 표현한 것이다. 실제 키와 파라미터의 생성 및 계산 과정보다 생성된 키와 파라미터의 유효성을 검증하는 과정에서 가장 많은 시간이 소모됨을 알 수 있다.

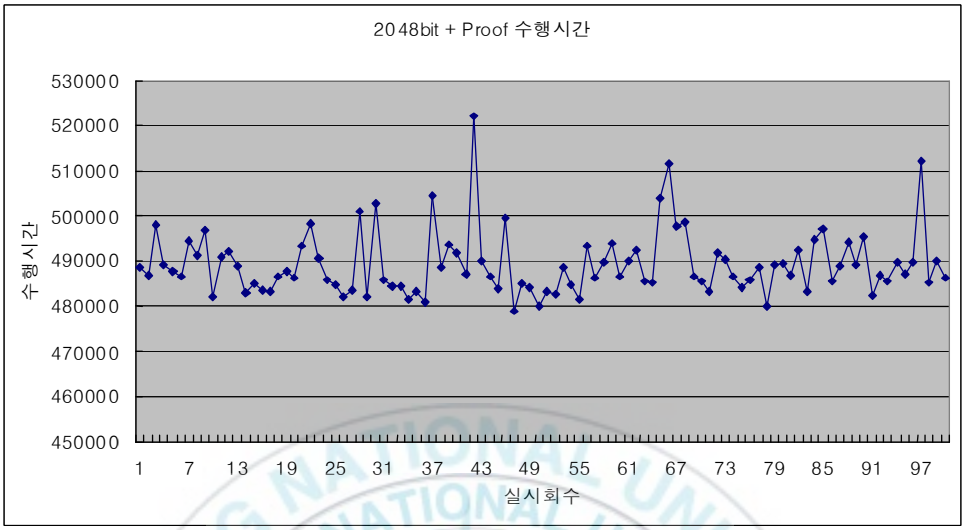

그림 20. 2048 bit에 대한 검증과정이 포함된 Setup 수행 시간(PC 기반)

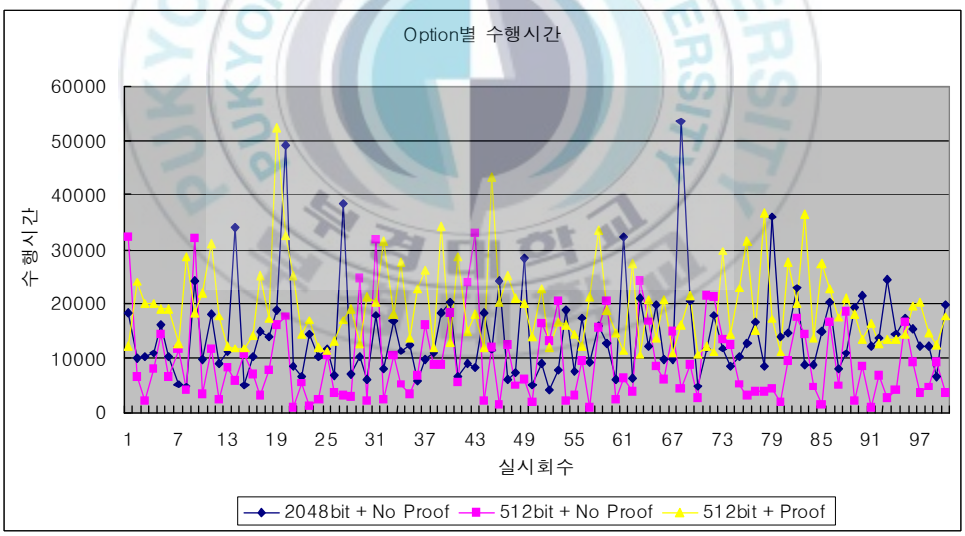

그림 21. 검증과정 포함여부에 따른 Setup 수행 시간(PC 기반)

기존의 DAA 프로토콜 Setup 과정을 HP iPaq Hx2790 포켓 PC와 IBM J9을 이용하여 수행한 결과는 표3과 같이 나왔다. 표 3은 PC 기반에서 512 bit의 검증과정이 미 포함된 Setup 과정의 시간과 PDA 기반에서

512 bit의 검증과정이 미 포함된 Setup 과정의 시간을 나타내는 결과이다.

표 3. PC 및 모바일 환경의 Setup 과정 수행 시간(단위:msec)

	PC 환경	PDA 환경
	512bit	512bit
최소소요시간	987	913,487
최대소요시간	32,875	4,141,472
평균소요시간	9,307	1,882,388

표 3에서 알 수 있듯이 기존의 PC환경에 맞게 구현된 DAA 프로토콜은 모바일 기기(PDA) 성능의 제약으로 인해 N 모듈러스의 크기가 512이고, 생성된 키에 대한 검증과정을 뺐음에도 불구하고 PC환경과 200배가 넘는 수행 시간을 볼 수 있다. 따라서 기존의 영지식 증명을 기반으로 하는 DAA 프로토콜은 PDA 및 모바일 환경에서는 사용이 불가능한 것으로 판단된다. 하지만 계산량을 많이 요구하는 Setup을 수행하는 발행자를 PC기반으로 만든 후, Join, Sign, Verify 과정만 모바일 환경에서 수행하게 구현한다면 모바일 환경에서도 충분히 DAA 프로토콜을 사용할 수 있을 것으로 판단된다. 그러나 실질적으로 모바일 환경에 DAA 프로토콜을 적용하기 위해서는 Setup에서 요구하는 계산량을 감소시키기 위한 경량화 및 새로운 프로토콜 제안이 필요하다는 것을 확인 할 수 있다.

4.2.2 제안된 새로운 DAA 프로토콜의 안전성 분석

DAA 프로토콜의 안정성을 분석하기 위해서는 DAA의 기반이 되는 그룹서명의 요구사항(2.2.2)을 만족하는 지를 살펴봐야 할 것이다. 제안된 DAA 프로토콜이 그룹 서명의 요구사항을 충족하면서 Strong RSA 가정을 만족한다면 이 프로토콜은 랜덤 오라클 모델에서 안전할 것이다. 제안

된 DAA 프로토콜은 다음과 같이 그룹서명의 요구사항을 만족한다.

- 건전성(Soundness)와 완전성(Completeness) : 공격자 E는 불법 TPM을 이용하여 생성한 인증서를 만들고자 발행자에게 Join 프로토콜을 요청하면, 발행자의 불법 TPM 검색(Rogue TPM)을 이용하여 불법 TPM인지를 판단하고 요청을 거부한다. 검증자도 마찬가지로 불법 TPM 검색을 이용하여 공격자 E의 요청을 거부한다.

- 위조 불가(Unforgeable) : TPM이 가지고 있는 EK의 비밀키를 기반으로 TPM의 인증서 및 서명이 생성을 생성한다. EK는 TPM이 만들어 질 때 TPM을 생산하는 공장에서 생성하여 넣는 내장된 비밀키이므로, 공격자 E는 절대 EK를 알아낼 수 없다. 따라서 공격자 E는 EK의 비밀키를 알지 못하면, TPM의 인증서 및 서명을 생성할 수 없다. 인증서 생성에는 Strong RSA 가정 및 이산대수 문제를 기반으로 하고 있어 공격자는 EK를 알지 못하면 인증서를 생성 할 수 없으며, 이로 인해 위조가 불가능 하다.

- 서명자 숨기기(Signer Ambiguous) : TPM에서 만들어지는 전자서명은 영지식 증명을 통해서 얻어진 값들을 이용하여 생성되고, 이를 검증할 때에도 영지식 증명을 통해서 이루어진다. 따라서 어떠한 메시지와 그 메시지의 서명이 주어지더라도 TPM의 인증서 생성에 필요한 비밀 정보 x가 공개 되지 않는다면, 영지식 증명으로 만들어진 메시지의 서명이 누구의 것인지 알기 어렵다. 따라서 공격자 E는 서명자의 신원을 파악할 수 없다.
- 서명 연결 불가(Uniteability) : 각 서명을 생성하는 단계에서 계산되

는 T_1, T_2의 값은 지식 서명을 통해 만들어 지고, TPM이 가지고 있는 비밀정보를 숨긴 상태에서 서명을 만든다. 따라서 공격자 E는 어떤 서명이라도 동일인이 서명을 했는지에 대해서 알기 어렵다.

- 공모 불가(No Framing) : DAA 프로토콜은 그룹을 관리하는 관리자를 가지고 있지 않아서 관리자와 그룹 소속원들이 공모하기가 어렵다. 또한 발행자와 검증자간의 공모를 통해 프라이버시 문제를 일으킬 수 있는 TPM 검색 방식에 대한 문제점을 본 논문에서는 수정하였으므로, 프라이버시에 대한 문제 발생도 차단시켰다. 따라서 공격자 E는 공모도 불가능하며, 공모에 대해 발생할 수 있는 문제점도 발생하기 어려울 것이다.

- 생성되지 않은 서명 처리 불가(Unforgeable tracing verification) : 검증자는 서명을 검증할 때 서명에 대한 검증을 수행하는 것이 아닌 TPM의 불법 유무를 판단하여 서명의 유효성을 판단한다. 따라서 유효한 TPM을 통해서 만들어진 서명이 아니라면 서명 처리를 할 수가 없다.

제안된 DAA 프로토콜은 영지식 증명을 통해서 이산 대수의 어려움을 기반으로 하고 있다. DAA 프로토콜을 통해서 생성되는 인증서 및 서명에 필요한 정보들은 공개되지 않은 채로 TPM과 발행자(또는 검증자)와 통신을 하기 때문에 공격자는 공격 대상인 TPM의 비밀 정보를 알기가 어렵다. 설령 공격자가 영지식 증명을 위해 만들어진 랜덤 수(s 또는 r)들을 알아낸다고 하더라도 랜덤 수들은 $s = r + cx$와 같이 비밀정보를 숨기기 때문에 공격에 성공할 수 없다.

4.2.3 제안된 새로운 DAA 프로토콜의 효율성 분석

이 절에서는 본 논문에서 제안하는 DAA 프로토콜의 효율성을 분석한다. 인증 스킴의 계산 복잡도는 모듈러 제곱과 곱셈의 연산에 의해서 좌우된다는 것을 알 수 있다. 계산 복잡도를 산출하기 위해서 본 논문에서는 Menezes, Oorschot 그리고 Vanstone[2]가 제시한 연산 방식을 이용할 것이다. 계산의 단순화를 위해서 일반적인 지수승 연산 기법에 기반을 둔 계산 복잡도를 산출한다. 일반적인 지수승 연산 기법은 다음과 같다.

- 특정 지수연산에 대해서 M_1을 지수의 이진비트 길이라고 두고, M_2를 이진수로 표현한 값들 중 1의 개수라고 두자. 예를 들어 이진수가 $(111101)_2$일 때 M_1은 6이고 M_2는 5가 된다.
- 우리는 M_1, M_2를 이용하여 M_1을 제곱 연산의 횟수(squaring)로, M_2를 곱셈(multiplication)을 계산 할 수 있다. 예를 들어, $y = g^x \bmod n$과 $x \in \{0,1\}^{160}$이 주어졌다고 가정하면 x를 이진수로 변경하고 나서 예상되는 1의 개수는 80(=160/2)이다. 따라서 예상되는 총 제곱 연산의 횟수는 160이고, 곱셈 수는 80이 된다.

DAA 프로토콜에서 TPM과 Host의 지수승 연산을 요구하는 프로토콜 과정은 시작(Join)과 서명(Sign) 프로토콜 과정이다. 따라서 TPM/Host에 대한 DAA의 계산 복잡도를 구하기 위해서는 시작(Join)과 서명(Sign) 과정에 대한 계산 복잡도를 구하면 될 것이다. 본 논문에서는 계산 복잡도를 비교하기 위해서 기존의 DAA 및 수정된 DAA 프로토콜의 총 지수 비트 길이를 계산해 보았다.

위의 일반적인 지수승 연산 기법을 사용하여 기존의 DAA 프로토콜과 본 논문에서 제안한 DAA 프로토콜의 연산 비교하면 표 4와 같다.

표 4. 기존의 DAA 프로토콜과 수정된 DAA 프로토콜의 비교

	기존의 DAA 프로토콜		수정된 DAA 프로토콜	
	제곱 연산 횟수	곱셈 횟수	제곱 연산 횟수	곱셈 횟수
준비(Join) 과정	6048	3,024	2,848	1,424
서명(Sign) 과정	25,566	12,783	5,770	2,885
합 계	31,614	15,807	8,618	4,309

표 4의 결과에 따라서 본 논문에서 제안하는 수정된 DAA 프로토콜은 기존의 DAA 프로토콜보다 3.6배가량 빠르다는 것을 알 수 있다. PC보다 덜 복잡한 기기에서도 DAA 프로토콜을 사용할 수 있는 기반을 마련해 줄 것이다.

또한 기존의 DAA 프로토콜은 많은 계산량을 처리하기 위해서 TPM과 Host에 연산 과정을 분산하여 처리를 하고 있다. 표 5를 통해서 알 수 있듯이 본 논문에서 제안하는 DAA 프로토콜은 기존의 DAA 프로토콜의 TPM이 가지는 연산량보다 적어 TPM 독자적으로 DAA의 전체 프로토콜을 수행할 수 있는 것이다. 이는 계산의 분산 없이도 DAA 프로토콜 수행이 가능하다는 것을 의미하며, 만약 계산의 분산을 Host와 TPM 사이에 나누어서 처리한다면 더욱더 빠른 연산을 가능하게 하여 모바일 디바이스와 같은 소형 디바이스에도 DAA 기술을 사용할 수 있을 것으로 예상된다.

표 5. TPM이 가지는 연산량 비교

	기존의 DAA 프로토콜		수정된 DAA 프로토콜	
	TPM	Host	TPM	Host
합 계	9,920	21,694	8,618	0

4.2.4 DAA 프로토콜의 수학적 기반

DAA 프로토콜은 영지식 증명의 수학적 기반을 이용하고 있다. DAA 프로토콜은 준비(Join), 검증(Verify)에서 영지식 증명을 이용하고 있다. 각각의 단계에서 사용된 영지식 증명의 수학적 기반의 타당성을 분석한다.

(1) 준비(Join) 단계

준비 단계에서 사용된 영지식 증명은 영지식 증명의 가장 기초적인 개념을 이용한 것이다. 영지식 증명의 기초적인 개념은 다음과 같다.

- 증명자와 검증자는 $ax = b \ mod \ n$이라는 것을 안다.
- 증명자는 r 값을 랜덤하게 선택해서 $B = a^r \ mod \ n$을 검증자에게 보낸다.
- 검증자는 c 값을 랜덤하게 선택해서 증명자에게 보낸다.
- 증명자는 $s = r+cx$를 계산하여 s 값을 증명자에게 보낸다.
- $a^s \equiv bB^c \ mod \ n$ 이라면 검증자는 증명자가 비밀 정보 x를 알고 있

다고 판단한다.

영지식 증명의 개념을 이용하여 본 논문의 준비(Join) 단계에서 사용된 영지식 증명을 하게 되면 다음과 같이 올바르게 영지식 증명을 수행한다는 것을 알 수 있다.

$$g^s \equiv g^{r+cx} \equiv g^r g^{cx} (\mathrm{mod} n)$$

$$\begin{aligned} BU^c &\equiv Bg^{xc} \bmod n \\ &\equiv g^r g^{xc} \bmod n \\ &\equiv g^{rxc} \bmod n \\ g^s &\equiv g^{(r+cx)} \bmod n \end{aligned}$$

$$\therefore BU^c \equiv g^s (\mathrm{mod} n)$$

BU^c와 g^s가 같으므로 $H(BU^c \| c)$와 $H(g^s \| c)$가 같은 값이 나오므로 발행자는 TPM이 비밀정보 x를 소유하고 있다는 것을 확신 하고 서명(Sign)에 필요한 v의 값을 생성하여 전달하게 된다.

(2) 검증(Verify) 단계

검증(Verify) 단계에서는 영지식 증명을 통해서 서명(Sign)단계에서 생성된 $\widetilde{T}_1, \widetilde{T}_2$을 서명에 공개된 정보들을 이용하여 검증하게 된다. $\widetilde{T}_1, \widetilde{T}_2$을 증명하기 위해서 사용되는 영지식 증명은 다음과 같다.

$$
\begin{aligned}
\widetilde{T_1} &:= T_1^{r_r} g^{r_x} \bmod n \\
&:= (r g^x S^v)^{r_r} g^{r_x} \bmod n \\
\widehat{T_1} &:= Z^{-c_h} T_1^{s_r} g^{s_x} \bmod n \\
&:= (T_1^r g^x)^{-c_h} T_1^{r_r + c_h r} g^{r_x + c_h x} \bmod n \\
&:= (g^x)^{-c_h} T_1^{r_r} g^{r_x + c_h x} \bmod n \\
&:= T_1^{r_r} g^{r_x} \bmod n \\
&:= (r g^x S^v)^{r_r} g^{r_x} \bmod n
\end{aligned}
$$

$$
\therefore \widetilde{T_1} = \widehat{T_1}
$$

$$
\begin{aligned}
\widetilde{T_2} &:= g^{r_w} \bmod n \\
\widehat{T_2} &:= T_2^{-c_h} g^{s_w} \bmod n \\
&:= (g^w)^{-c_h} g^{r_w + c_h w} \bmod n \\
&:= g^{r_w} \bmod n
\end{aligned}
$$

$$
\therefore \widetilde{T_2} = \widehat{T_2}
$$

검증 과정에서는 T_1, T_2를 생성할 때 사용된 x, v, w 값을 검증자가 알지 못하더라도 영지식 증명을 통해서 x, v, w 값을 알고 있다는 것을 검증자는 확신하게 된다. 따라서 검증자는 검증을 요청하는 TPM이 유효한 비밀 정보 x와 발행자로부터 받은 v를 가지고 있다는 것이 증명되었으므로 검증을 완료할 수 있게 된다.

5. 결론

많은 대형 IT 기업들이 참여하고 있는 TCG는 기존의 소프트웨어 보안 기술의 문제점을 해결하기 위해서 하드웨어 기반의 보안 기능을 제안하였다. TCG에 의해 제안된 TPM과 DAA는 유비쿼터스 환경에서 신뢰 컴퓨팅을 제공하기 위해서 필수적인 항목이 될 것으로 예상된다.

본 논문에서는 신뢰 컴퓨팅과 원격 인증을 제공하기 위해서 TCG에서 제안한 DAA에 대해서 분석하고 DAA에 대한 문제점을 파악하였다. 또한 DAA의 문제점을 개선시킨 수정된 DAA 프로토콜 및 DAA가 가지고 있던 프라이버시 문제에 대한 문제점을 해결할 수 있는 해결책을 제시하였다.

영지식 증명, Camenisch와 Lysyanskaya의 서명 스킴과 Camenisch와 Stadler의 서명 스킴의 장점을 살려 제안된 새로운 DAA 프로토콜은 많은 계산량을 요구하여 PC 기반이 아닌 다른 디바이스 환경에서 사용이 어려웠던 기존의 DAA 프로토콜이 가지던 문제점을 해결하였다. 또한 DAA가 가지고 있던 프라이버시 문제점을 해결하여 DAA의 기존 목적인 프라이버시 보호 및 보안 기능 제공 기능을 한층 강화 하였다. 제안된 DAA 프로토콜은 기존의 프로토콜에 비해 3배 이상 빨라서 유비쿼터스 환경 및 모바일 환경에서 DAA의 적용을 가능하게 해줄 것이다.

또한 4장에서는 TCG에서 제안하고 있는 DAA 프로토콜과 수정된 DAA 프로토콜의 구현 결과를 보여주었다. IBM의 DAA 프로토콜을 상용으로 제공되고 있어 DAA 프로토콜을 연구하는데 어려움이 있었던 것이 사실이다. 그러나 본 논문에서 설계한 프로토콜을 이용한다면 이러한 어려움이 해결될 것이며, 이로 인해 DAA에 대한 연구가 더욱더 활발해 질

것으로 예상된다.

제안된 프로토콜은 현재 Java 기반으로 구현되었으며, 현재로는 다른 언어와의 호환이 불안정하다. 또한 TPM과의 100% 호환이 되지 못하는 문제점을 지니고 있다. 그리고 TCG에서 제안하는 TSP, TDDL 라이브러리 등에 맞는 C 인터페이스를 제공하지 못해 TCG에서 제안하는 TPM의 전반적인 환경에 대한 테스트가 불가능하다. 따라서 향후에는 실무 환경에서 실행 가능한 C 인터페이스 기반의 DAA 프로토콜를 추가로 구현 할 것이며, 현재 지니고 있는 DAA의 문제점을 단계적으로 해결해 나갈 것이다.

참고문헌

[1] A. Lysyanskaya, “Signature schemes and applications to crypto graphic protocol design,” PhD thesis, Massachusetts Institute of Technology, Cambridge, massachusetts, Sept. 2002

[2] A. J. Menezes, P. C. Oorschot, and S. A. Vanstone. Handbook of Applied Cryptography, CRC press, Inc, 1997.

[3] C. Popescu, “An efficient id-based group signature scheme,” Studia Universitatis Babes-Bolyai, Informatica, Vol. XLVII, November 2, 2002, pp. 29-36.

[4] Dan Bonech, Xavier Boyen, Hovav Shacham, “Short Group Signatures,” In proceedings of Crypto 04, LNCS 3152, 2004, pp. 41-55.

[5] David Chaum, Eugene van Heyst, "Group Signature, Advances in Cryptography," Euro-CRYPT 91, 1991, pp. 257-265.

[6] E. Brickell, J. Camenisch, and L. Chen, "Direct anonymous attestation," In Proceedings of 11th ACM Conference on Computer and Communications Security, ACM Press, 2004, pp. 132-145.

[7] Fiat and Adi Shamir, "How To Prove Yourself: Practical Solutions to Identification and Signature Problems," Advances in Cryptology : Proceedings of Crypto 86, LNCS, 1987, pp. 186-194

[8] G. Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik, "A Practical and Provably Secure Coalition-Resistant Group Signature Scheme," Advances in Cryptography , CRYPTO 2000, 2000, pp. 255 - 270

[9] He ge, "A Method to Implement Direct Anonymous Attestation,"

2006, http://citeseer.ist.psu.edu/ge06method.html

[10] J. Camenisch, "Direct Anonymous Attestation : Achieving Privacy in Remote Authentication," ISC2004, 2004.

[11] J. Camenisch, M. Stadler. "Efficient Group Signature Schemes for Large group," Advances in Cryptology - CRYPTO '97, LNCS 1294, 1997, pp. 410-424.

[12] J. Camenisch, A. Lysyanskaya. "A signature scheme with efficient protocols," Security in Communication Networks, Third International conference, SCN 2002, LNCS 2576, 2003, pp. 268-289.

[13] Ronald L. Rivest, "RSA Problem," Encyclopedia of Cryptography and Security, 2003, pp. 501-538.

[14] L. Chen and T. P. Pedersen, "New Group Signature Schemes," Advances in Cryptology Eurocrypt 1994, LNCS 950, 1994, pp. 171-181.

[15] Mobile Platforms Group intel Corporation, Trusted Platform Module
(TPM) based Security on Notebook PCs, White Paper, 2002

[16] Ronald Cramer. "Signature schemes based on the Strong RSA Assumption," ACM Transactions on Information and System Security (ACM TISSEC), 2000, pp. 161-185.

[17] Smyth, B., Ryan, M. & Chen, L. "Direct Anonymous Attestation (DAA): Ensuring privacy with corrupt administrators." In proceedings of the Fourth European Workshop on Security and Privacy in Ad hoc and Sensor Networks, LNCS 4572, 2007, pp. 218-231.

[18] TCG, "Mobile Trusted Module Specification Overview Document,"

Specification, 2006

[19] TCG, "Mobile Trusted Module Ecosystem Graph," White Paper, 2006

[20] TCG, "Mobile Phone Work Group Use Cases," White Paper, 2005

[21] TCG, "TCG Specification Architecture Overview Specification Revision 1.3," Specification, 2007

[22] TCG, "TPM Main Part 1-3 Specification Version 1.2," Specification, 2007

[23] TCG, "TCG Software Stack (TSS) Specification Version 1.2 Level 1," Specification, 2006

[24] Xiaofeng Chen, Fangguo Zhang and Kwangjo Kim, "A New ID-based Group Signature Scheme from Bilinear Pairings," Cryptology ePrint Archive, LNCS 3348, 2003, pp. 371-383.

[25] 권성구, TPM을 이용한 에이전트 기반의 안전한 네트워크 자가형성 기법 연구, 중앙대학교 석사 졸업 논문, 2006

[26] 김영수 외, "신뢰 컴퓨팅과 TCG 동향," 전자통신동향분석 제22권 제1호, 2007, pp. 83-96.

[27] 박우람, 박찬익, "TPM과 네트워크 스토리지" 보안주간기술 동향 통권 1279호, 2007, pp. 17-27.

[28] 주학수, 김대엽, 이동훈, "그룹서명을 이용하여 익명성이 보장되는 디지털 권한 전달 시스템," 정보보호학회 논문지, 제 14권, 제 1호, 2004, pp. 3-7

[29] 장기식, 보안을 위한 효율적인 방법 PKI, 인포북, 2003.

[30] 이기열 외, 클러스터 기반의 이동 Ad Hoc 네트워크에 DAA 기술을 통한 신뢰할 수 있는 인증 모델 설계, 한국정보과학회 추계 학술대회

논문집 A, 2007