情 報 保 護 學 會 誌
第 14 卷 第 6 號, 2004. 12

공개키 기반 구조에 기반한 익명게시판 기술 현황

권 태 경*, 박 해 룡**, 이 철 수***

요 약

인터넷 게시판에서 실명을 사용할 경우 자유로운 토론이 어려우며 사용자 프라이버시를 침해할 우려가 있는 반면, 가명을 사용할 경우 자유로운 토론은 가능하지만 오히려 상호 비방이나 유언비어 등의 부작용이 있을 수 있다. 따라서 기본적으로는 가명을 이용해서 포스팅하도록 허용하지만, 필요한 경우 분산된 여러 개체간의 합의에 의해서 조건부 실명 복원(혹은 다른 말로 조건부 추적)이 가능한 게시판이 구현된다면 매우 유용할 것이다. 그러나 기존 체계에서 가명만을 이용하여 조건부 추적 가능한 익명성을 제공하기란 쉽지 않다. 또한 현존하는 익명성 제공 기법들을 기존의 인증 체계나 인증서 체계에서 수용하기는 매우 어렵다. 본 논문에서는 이와 같이 인터넷 게시판에서 익명성을 제공할 수 있는 기술들을 간략히 살펴보고, 특히 기존의 공개키기반구조, 즉 X.509 인증서 체계를 이용하여 익명게시판을 구현할 수 있는 기술에 대해서 소개하도록 한다.

Ⅰ. 서 론

최근 인터넷 게시판에서는 기본적으로 주민등록번호를 포함한 개인정보를 통해서 실명 등록을 한 후, 해당 계정으로 로그인 후 게시판을 사용하도록 권장하는 경우가 늘고 있다. 이것은 인터넷 게시판에서 실명을 사용할 경우 자유로운 토론이 어려우며 결과적으로 사용자 프라이버시를 침해할 우려가 있는 반면, 가명을 사용할 경우 자유로운 토론은 가능하지만 오히려 상호 비방이나 유언비어 등의 큰 부작용이 있을 수 있기 때문이다. 게시판을 제공하는 입장에서 후자의 경우가 더 큰 폐해로 나타나면서 오히려 인터넷 게시판은 사용자 프라이버시를 경시하게 되는 사례가 늘어가고 있다. 따라서 기본적으로는 가명을 이용해서 포스팅하도록 허용하지만, 필요한 경우 분산된 여러 개체간의 합의에 의해서 조건부 실명 복원(혹은 다른 말로 조건부 추적)이 가능한 게시판이 구현된다면 매우 유용할 것이다. 그러나 기존 체계에서 가명만을 이용하여 조건부 추적 가능한 익명성을 제공하기란 쉽지 않다. 또한 현존하는 익명성 제공 기법들을 기존의 인증 체계나 인증서 체계에서 수용하기는 매우 어렵다. 만약 전자서명의 조건부 추적 가능한 익명성 제공을 위한 기술이, 기존 공개키기반구조 체계에서 별다른 수정없이 활용 가능하다면 보다 쉽게 구현되고 사용될 수 있을 것이다. 현재 국내·외에서 사용되는 공개키 인증서의 경우 소유자의 실명을 포함하도록 규정하고 있다. 은행 거래, 증명서 발급 등 실명이 필요한 경우에는 이러한 공인인증서가 유용하지만, 조건부 추적 가능한 익명 게시판에서 사용되기 위해서는 부가적인 기술이 요구된다. 본 논문에서는 이와 같이 인터넷 게시판에서 익명성을 제공할 수 있는 기술들을 간략히 살펴보고, 특히 기존의 공개키기반구조, 즉 X.509 인증서 체계를 이용하여 익명게시판을 구현할 수 있는 기술에 대해서 소개하도록 한다. 따라서 먼저 II장에서는 게시판과 같은 환경에서 익명성을 제공하기 위한 세부 기술들을 살펴보고, III장에서는 X.509 인증서 체계를 이용한 기술을 소개한다. 그리고 IV장에서는 익명게시판 구현 사례를 소개하고, V장에서 결론을 맺는다.

Ⅱ. 기존 기법 분석 및 요구사항

사용자 익명성을 갖는 전자서명 인증 기술은 1985년 D. Chaum에 의해 제안되었으나, 최근 사용자의 프라

* 세종대학교 컴퓨터공학부 (tkwon@sejong.ac.kr)
** 한국정보보호진흥원 암호인증기술팀 (hrpark@kisa.or.kr)
*** 경원대학교 소프트웨어대학 (csl100@kyungwon.ac.kr)

이버시 보호 측면이 대두됨에 따라 다시 활발한 연구가 진행되고 있다. 이 장에서는 익명성을 갖는 전자서명 인증 기술의 제안에서부터 최근 발표된 기술까지 전체 연구 동향을 살펴보고, 최근 제안된 세 가지 인증 기술에 대하여 구체적으로 알아본다.

1. 연구 동향

서비스 제공자에 대한 사용자의 익명성을 유지하며 신분을 인증하는 신용장(credential)에 대한 연구는 1985년 D. Chaum에 의해 제안되었다[4]. 디지털 신용장은 개인과 관련하여 기관에 의해 발행되고, 다른 기관에 보일 수 있는 문서라고 할 수 있다. 이후 1987년 D. Chaum과 J. Evertse는 모든 트랜잭션에 관여하는 준신뢰 기관의 존재를 기반으로 한 해결책을 제시하였으나, 이러한 준신뢰 기관의 존재는 구현 효율적으로나 안전성 측면 양면에서 바람직하지 않다[5].

이후 Damgard에 의해 제안된 방법은 계산복잡도 이론에 근거한 일방향 함수와 영지식 증명을 사용하여 실제 적용하기에는 어려움이 있고, 사용자들의 결탁에 대한 보호 방법이 제시되어 있지 않다[6]. Chen은 1995년 이산대수에 기반한 은닉서명을 이용한 해결책을 제시하였다. 이 방법은 효율적이기는 하지만, 사용자들의 결탁에 대한 대비가 없고 신용장을 여러 번 사용하기 위해서는 발행기관으로부터 여러 번 발급 받아야만 한다[7].

Lysyanskaya 등은 1999년 일반적인 신용장 시스템을 제안하였다. 이 시스템은 필요한 요구사항의 많은 부분을 제대로 해결하고 있지만, 일방향 함수와 일반 영지식 증명에 기반하고 있기 때문에 실제로 적용하기에는 어려움이 있다[8]. Lysyanskaya 등이 제안한 비표준적 이산대수 문제에 기반한 예제는 Chen이 제안한 방법과 마찬가지로 신용장의 사용횟수가 제한적이라는 문제를 가지고 있다.

2. 기존 방식의 기능비교

Brands 방식의 가장 큰 특징은 사용자가 디지털 신용장의 여러 속성 중 일부를 선택적으로 공개할 수 있다는 것이다[1]. 예를 들어, 사용자는 자신이 자녀를 양육하고 있다는 사실은 공개하면서, 자녀의 수와 나이 등은 비밀로 할 수 있다. 이러한 공개 속성들이 옳다는 것은 확인자의 nonce에 대한 사용자의 전자서명을 통해 증명된다.

디지털 신용장의 추적불가능성(untraceability)의 보장을 위해 초기 등록부터 익명 채널을 이용한다는 것은 현실적으로 바람직하지 않다. 사용자는 디지털 신용장을 사용할 때, 신용장의 공개키와 발급기관의 서명을 제시하기 때문에, 발급기관은 이 두 데이터를 신용장 발급단계에서 볼 수 없어야 한다. 만일 발급기관이 데이터를 볼 수 있다면, 확인자와 결탁하여 사용자의 신분을 추적할 수 있다. 이와 함께, 사용자가 발급기관이 인코딩 하고자 하는 모든 속성들을 은닉할 수 없어야 한다. 사용자가 임의로 모든 속성들을 조작할 경우, 자신에게 불리한 정보를 임의대로 편집할 수 있기 때문이다. 이 두 성질을 만족하기 위해 Brands 방식에서는 제한적 은닉(restrictive blinding)이라고 불리는 기술을 사용한다.

사용자가 신용장을 한번 이상 사용할 경우, 제시한 신용장들로부터 같은 사용자가 접속했다는 정보를 얻을 수 있어서는 안 된다. 이러한 이유로 같은 신용장이 너무 많이 사용되는 것은 바람직하지 않고, 일회용으로 제한하는 것이 안전도 면에서는 가장 이상적이다.

또한 선택적인 기능으로 확인자는 신용장 제시 프로토콜의 기록(transcript)을 신뢰기관에 저장하여, 신뢰기관이 신용장의 위조나 사용회수 제한 등을 확인할 수 있도록 할 수 있다.

Brands 방식에서 제공하고 있는 특징들을 정리해 보면 다음과 같다.

- o 인증성(Authenticity)
- o 익명성(Anonymity)
- o 결합익명성(Unlinkability)
- o 위조 방지(Protection against forgery)
- o 공유 방지(Protection against sharing)
- o 사용회수 제한(Limit show)
- o 추적성(Traceability)
- o 선택적 은닉(Divisibility)

Camenisch-Lysyanskaya 방식은 우선 사용자의 신용장에 대한 위조를 할 수 없도록 하고 있다[2]. 각 익명과 신용장은 잘 정의된 사용자에게 속해야 하며, 여러 사용자들이 협력을 해도 한 사용자의 정보를 유추할 수 없어야 한다. 또한, 사용자의 프라이버시를 제공해야 한다. 사용자의 신용장을 사용자가 소유하고 있다는 점을 제외하고 다른 사항들을 알아 낼 수 없어야 한다. 한 사용자가 여러 익명을 사용한다 해도, 그 익명들을 통해 사용자를 유추할 수 없어야 한다. 그리고, 시스템은 전체적으로 효율적이어야 한다.

추가적인 성질로는 사용자들이 자신의 익명이나 신용장을 남들과 공유하지 못하도록 해야 한다. Camenisch-Lysyanskaya는 all-or-nothing 전송가능성이라는 개념을 이용하여, 사용자가 자신의 신용장의 일부를 남과 공유하게 될 경우 자신의 전체 비밀 정보를 노출하게 되도록 구현하고 있다. 또한 비인가된 트랜잭션에 대해 사용자의 신원을 알아낼 수 있는 기능과 사용회수를 제한한 일회용 신용장을 제안하고 있다

Camenisch-Lysyanskaya 방식이 제공하는 성질들을 정리하면 다음과 같다.

- o 인증성(Authenticity)
- o 익명성(Anonymity)
- o 연결 익명성(Unlinkability)
- o 위조 방지(Protection against forgery)
- o 공유 방지(Protection against sharing)
- o 사용회수 제한(Limit show)
- o 추적성(Traceability)
- o 효율성(Efficiency)

Verheul 방식은 기본적으로 D. Chaum이 제안한 여러 익명에서 사용자를 식별할 수 없도록 하는 연결 익명성(unlinkability)와 사용자가 익명을 전환할 수 있는 translatability를 제공하고 있다[3]. 이 기본 특성에 추가하여, 신용장의 위조 방지, 신용장 공유 방지, 인증서 폐기 기능을 제안하고 있다

- o 인증성(Authenticity)
- o 익명성(Anonymity)
- o 연결 익명성(Unlinkability)
- o 인증서 전달성(Translatability)
- o 자기 은닉성(Self-blindability)
- o 위조 방지(Protection against forgery)
- o 공유 방지(Protection against sharing)
- o 사용회수 제한(Limit show)
- o 추적성(Traceability)
- o 효율성(Efficiency)

3. 요구 사항

안전한 익명 인증서 시스템 구축을 위해서는 다음과 같은 공통 요구조건이 필요하다.

o 익명 요구 조건

- 익명성(Anonymity): 사용자의 익명 인증서으로부터 사용자에 대한 정보를 유추할 수 없어야 한다.
- 결합익명성(Linkability): 사용자가 여러 익명 인증서를 사용해도 인증서를 통해, 사용자에 대한 정보를 유추할 수 없어야 한다.

o 안전성 요구 조건

- 인증성(Authenticity): 인증서를 이용하여 사용자를 정확히 인증할 수있어야 한다.
- 책임성(Accountability): 인증서와 사용자는 올바르게 연결되어야 한다.
- 위조 방지(Protection against forgery): 여러 사용자가 결탁하여도 사용자의 인증서를 위조할 수 없어야 한다.
- 공유 방지(Protection against sharing): 사용자가 자신의 인증서를 타인과 공유 또는 양도할 수 없어야 한다.
- 무결성(Integrity): 인증서에 대한 무단 변조나 변경이 불가능해야 한다.

o 추적 및 폐기 요구 조건

- 추적성(Traceability): 비합법적인 사용과 같은 특정 조건 하에서 사용자의 실제 신분을 추적할 수 있어야 한다.
- 폐기성(Revocation): 인증서의 비밀키 노출과 같은 특정 조건 하에서인증서의 기능을 폐기할 수 있어야 한다.

o 효율성 요구 조건

전체적인 시스템은 효율적으로 운용될 수 있어야 하고, 중앙 집중에 따른 병목현상을 방지하여 서비스 거부 공격에 내성을 가져야 한다.

이러한 기본 요구 조건 이외에 선택적인 요구 조건으로는 다음과 같은 사항들이 있다.

o 사용회수의 제한

필요에 따라 인증서의 사용 회수를 제한할 수 있어야 한다.

- 일회성(One-time show): 인증서의 이중사용을 방지하여, 한번만 사용하도록 할 수 있어야 한다.
- 유일성(Unicity): 각 사용자는 인증서를 오직 한 번만 발급받을 수 있어야 한다.

o 편의성

- 선택적 은닉(Divisibility): 사용자들이 인증서의

여러 속성들을 선택적으로 공개할 수 있어야 한다.
- 결합성(Combination): 사용자들이 여러 가지 인증서를 조합, 결합하여 하나의 통합된 인증서를 생성할 수 있어야 한다.
- 재증명성(Translatability): 인증서의 수신자가 제 3자에게 그 사실을증명할 수 있어야 한다.
- 자기 은닉성(Self-Blindability): 기존의 익명 인증서로부터 사용자 스스로 새 인증서를 유도할 수 있어야 한다.

Ⅲ. 익명 인증서 프로토콜

1. 프로토콜 설계 개념

1.1 설계 요구사항

기존의 공개키기반구조(PKI: Public Key Infrastructure)를 바탕으로 운영될 수 있는 익명 인증서 및 인증서 프로토콜을 소개한다[9]. 이것은 현재 국내에서 주로 사용되고 있는 RSA 전자서명 알고리즘을 기반으로 동작할 수 있는 프로토콜을 구성의 기본 원칙으로 삼으며, 현재 관련 논문을 작성하여 제출 중이다. 이와 같이 익명 인증서 시스템을 구축할 경우, 기존 체계의 큰 변화 없이 개인정보보호를 위한 익명 인증서 서비스를 수용할 수 있다는 장점이 있다. 즉, 프로토콜 설계를 위한 기본적인 요구사항을 다음과 같이 구성할 수 있다.

- o 기존 공개키기반구조와의 호환성 또는 상호운용성 제공
- o RSA 전자서명 알고리즘 지원
- o 가명을 통한 익명 인증서의 기본적인 기능과 응용 목적 지원

익명 인증서의 기본적인 기능으로는 앞에서 설명한 바와 같이 익명성 유지, 인증, 계정 관리, 다중 발급, 위변조 방지, 추적성, 인증서 공유 방지, 취소, 다중 사용, 그리고 신용 증명 등이 있다. 한편 구체적인 응용 목적으로는 일반 웹 사이트와 같이 범용성이 강한 응용, 약국 처방전과 같이 일회성 등 그 사용 제한이 강한 응용, 그리고 투표와 같이 사용 제한뿐만 아니라 정보 은닉성을 부가적으로 요구하는 응용 등으로 나누어볼 수 있다.

1.2 프로토콜의 표기법

프로토콜의 가독성을 높이기 위하여 먼저 다음과 같이 표기법을 정의한다.

U : 사용자

AI : (익명 인증서) 발급자 1 - 익명 발급자

BI : (익명 인증서) 발급자 2 - 은닉 발급자

PI : (익명 인증서) 발급자 (AI, BI)

CA : (기존) 공인인증기관

SP : 응용 서비스 사이트

ID_U : 사용자의 실명 또는 해당 ID

CT_U : 사용자의 공인인증서

PN_U : 사용자의 익명

pk_X : X의 공개키

sk_X : X의 개인키

ppk_U : 사용자의 익명 공개키

psk_U : 사용자의 익명 개인키

SIG_X : X의 개인키에 의한 전자서명 기능

ENC_X : X의 공개키 혹인 비밀키에 의한 암호 기능

$H()$: 강한 일방향 해쉬 함수 또는 이를 포함하는 인코딩 기법

$\leftarrow_R$: 난수 선택

e : PI의 공개 지수

d : PI의 개인 지수

d_1 : BI의 개인 지수

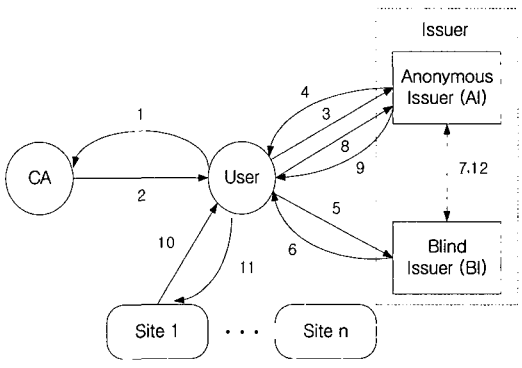
d_2 : AI의 개인 지수

N : 발급자의 RSA 법

r : 사용자의 은닉 인수

M : 익명 인증서의 내용

SN : 익명 인증서의 일련번호



〔그림 1〕 프로토콜 흐름도

b : 익명 인증서의 헤더

c_i : 크리덴셜

κ, ℓ : 암호 변수

이와 같이 익명 인증서의 서명 방식을 구체적으로 다루기 위하여, 프로토콜을 기술할 때 발급자의 서명 알고리즘만 구체적인 RSA 파라메타를 사용하여 표기하도록 하며, 그 외의 전자서명은 SIG_X와 같이 간략히 나타내도록 한다. 특히 편의상 $\{M\}_{SIG_X}$와 같이 표기할 경우 원본 메시지 M과 서명값 $\{M\}_{SIG_X}$가 함께 존재하는 것으로 가정한다.

2. 익명 프로토콜

2.1 기본 모델

익명 프로토콜의 구성 모델은 다음의 [그림 1]과 같다. 가장 기본적인 아이디어는 익명 인증서 발급자를 AI (Anonymous Issuer: 익명발급자)와 BI (Blind Issuer: 은닉발급자)로 구분하여, 익명발급자는 내용에 대한 확인만 할 수 있으며 은닉발급자는 신원에 대한 확인만 할 수 있도록 구성하는 것이다. 즉, 익명발급자는 신원에 대한 확인을 할 수 없으므로, 또한 은닉발급자는 내용에 대한 확인을 할 수 없으므로, 서로 합의하지 않고는 사용자에 대한 신원 추적이 불가능 하도록 한다. 이와 같이 신원 추적 기능을 제공하면서 내용에 대한 확인이 가능할 수 있는 모델이다. 하지만 전자투표와 같이 제한적 신원추적을 위해서 자체 은닉(Self-Blinding)을 활용할 수도 있다.

o 단계 1과 2

익명 인증서를 발급받기 위하여 사용자는 반드시 공인인증기관으로부터 기존의 공인인증서를 발급받아야 한다.

o 단계 3과 4 (플로우 생략 가능)

익명 인증서를 발급받기 위해서 사용자는 익명발급자에게 인증서 스켈레턴(skeleton)을 요구하거나, 주어진 클라이언트 응용 환경에서 직접 만들도록 한다. 스켈레턴을 요구할 경우 실명을 사용하지 않으며, 이때 익명발급자는 자신의 발급 양식에 맞게 익명 인증서의 기본 포맷인 스켈레턴을 발급한다. 사용자가 직접 만들 경우에는 익명발급자의 발급체계를 준수하도록 하며, 예를 들면 충분한 크기의 랜덤값을 일련번호로 사용한다. 결과적으로 단계 3과 4의 메시지 플로우는 제거할 수 있다.

o 단계 5와 6

공인인증서를 바탕으로 은닉발급자로부터 익명 인증서의 부분 서명을 받을 수 있다. 먼저 사용자는 스켈레턴을 바탕으로 익명 인증서의 내용에 해당하는 값 $M=\langle PN_U, ppk_U, INFO_ATTR\rangle$을 구성해야한다. $INFO_ATTR$항목에 익명하에 필요한 추가적인 정보를 포함하여 그 응용 범의를 넓힐 수 있다. 이때 은닉발급자는 공인인증서를 바탕으로 사용자의 신원만을 파악하며, 익명인증서의 내용을 열람할 수는 없다. 은닉발급자는 향후 분쟁 발생시 추적을 위한 기록을 남긴다. 단계 5와 6은 비밀성이 보장된 암호채널이어야 한다.

o 단계 7와 12

익명발급자와 은닉발급자는 서로 제어정보를 주고받을 수 있다. 단계 7과 12는 비밀성이 보장된 암호채널이어야 한다.

o 단계 8과 9

사용자는 은닉발급자로부터 받은 부분 서명을 완성하기 위하여, 익명발급자에게 나머지 부분서명을 요구하도록 한다. 이때 익명발급자는 익명인증서의 내용을 열람하고 검증할 수 있으며, 사용자의 신원을 파악할 수는 없다. 익명발급자는 향후 분쟁 발생시 추적을 위한 기록을 남긴다. 단계 8과 9는 비밀성이 보장된 암호채널이어야 한다.

o 단계 10과 11

익명 인증서를 발급 받은 사용자는 다양한 목적에 따라서 서비스 제공 사이트에서 인증서를 사용할 수 있다. 본 프로토콜에서는 기본적으로는 익명 인증서를 이용한 challenge-response 인증 방식을 가정한다. 하지만 그 외에도 메시지 등록 등 다양한 목적으로 익명 인증서를 사용할 수 있다. 예를 들면, (게시판 서버와 같은) 해당 서비스 서버에 초기 등록 시 서버는 사용자가 제시한 익명 인증서와 서명을 통해서 사용자를 인증하며, 인증된 사용자는 원하는 아이디와 패스워드를 선택·등록하도록 한다. 이어서 서버는 해당 사용자의 [아이디, 패스워드 검증 정보, 익명 인증서] 프로파일을 구성하고 저장한다. 이 후 사용자는 일반적인 경우와 마찬가지로 아이디와 패스워드만을 이용해서 서비스를 사용할 수 있으며, 서버는 해당 사용자의 실명 복원을 원하는 경우 사용자의 가명 아이디가 아닌 익명 인증서를 이용하여 AI와 BI에게 실명 복원을 요청할 수 있다.

2.2 프로토콜 구성

이어서 추적가능 프로토콜 (Traceable Protocol)을 소개한다. 프로토콜은 [그림 1]의 기본 모델을 따르며, 따라서 익명 인증서 발급자를 익명발급자와 은닉발급자로 나누도록 한다. 특히 익명발급자와 은닉발급자가 다중서명을 통해서 익명인증서에 대한 발급자로서의 서명을 해야하는데, 이것을 RSA 다중서명을 통해서 해결하도록 한다.

RSA 다중서명을 위해서 익명 인증서 발급자의 개인키 $\langle d,N\rangle$은 $\langle d_1,N\rangle$과 $\langle d_2,N\rangle$으로 나누어서, 익명발급자와 은닉발급자가 나누어 갖도록 한다. 이 때 $d=d_1d_2\phi(N)$이며, 은닉발급자는 $\langle d_1,N\rangle$을 익명발급자는 $\langle d_2,N\rangle$을 소지하여 서명에 이용하도록 한다. 각 발급자의 서명을 부분서명이라 부른다.

2.3 추적가능 프로토콜

본 논문에서 소개하는 프로토콜의 추적가능 버전은 [그림 2]와 같이 구성된다.

(1) $U\Rightarrow BI$: $\left\{\{u\}_{SIG_U},\rho\right\}_{ENC_{BI}}, CT_U$

U는 X.509 인증서 스켈레톤을 생성하고 $b\leftarrow\langle PN_U, ppk_U, SIG_U\rangle$와 $M\leftarrow\langle b,(c_i)\rangle$를 정의한다. U는 $SN=H(PI,PN,ppk_U,SIG_{PN})$과 $h=H(M)$을 계산하고 $u=h\cdot r^e \bmod N (r\leftarrow_R\{0,1\}^\kappa)$를 계산한다. 최종적으로 $\left\{\{u\}_{SIG_U},\rho\right\}_{ENC_{BI}}$ $(\rho\leftarrow_R\{0,1\}^\ell)$를 계산하여 사용자 공인인증서 CT_U와 함께 BI에게 전송한다. ρ는 U와 BI 사이의 세션키로 사용된다.

BI는 pk_{CA}를 사용하여 CT_U를 검증한다. BI는 $\left\{\{u\}_{SIG_U},\rho\right\}_{ENC_{BI}}$를 복호하고 pk_U를 사용하여 $\{u\}_{SIG_U}$를 검증한다. BI는 $w=u^{d_1} \bmod N$를 계산하고 $\langle\{u\}_{ENC_{BI}}:ID_U\rangle$을 저장장치에 기록한다. 이것은 향후 분쟁 발생시 추적을 위해서 사용될 값이다. 마지막으로 $\{w\}_{ENC_{AI}}\oplus\rho$를 계산하여 U에게 돌려준다.

(2) $BI\Rightarrow U$: $\{w\}_{ENC_{AI}}\oplus\rho$

U는 ρ를 사용하여 $\{w\}_{ENC_{AI}}\oplus\rho$에서 $\{w\}_{ENC_{AI}}$를 밝혀내고 $\left\{\{M\}_{SIG_{PN}},r,\{w\}_{ENC_{AI}}\right\}_{ENC_r}$를 계산하여 AI에게 전송한다. $\{M\}_{SIG_{PN}}$는 sk_U가 아닌 psk_U를 사용한 서명임에 주의하라.

(3) $U\Rightarrow AI$: $\left\{\{M\}_{SIG_{PN}},r,\{w\}_{ENC_{AI}}\right\}_{ENC_r}$

AI는 ppk_U를 사용하여 $\{M\}_{SIG_{PN}}$를 검증하고 $z=w^{d_2}$

(1) $U\Rightarrow BI$: $\{\{u\}_{SIG_U},\rho,CT_U\}_{ENC_{BI}}$ · · · [그림 1]의 단계 5에 해당
(*) U defines $b\leftarrow\langle PN_U, ppk_U, SIG_U\rangle$ and $M\leftarrow\langle b,(c_i)\rangle$
(*) U computes $SN=H(PI,PN,ppk_U,SIG_{PN})$ and $h=H(M)$
(*) U computes $u=h\cdot r^e \bmod N$ where $r\leftarrow_R\{0,1\}^\kappa$
(*) U computes $\{\{u\}_{SIG_U},\rho\}_{ENC_{BI}}$ where $\rho\leftarrow_R\{0,1\}^\ell$
(*) BI computes $w=u^{d_1} \bmod N$
(*) BI records $\langle\{u\}_{ENC_{BI}}:ID_U\rangle$
(*) BI computes $\{w\}_{ENC_{AI}}\oplus\rho$
(2) $BI\Rightarrow U$: $\{w\}_{ENC_{AI}}\oplus\rho$ · · · [그림 1]의 단계 6에 해당
(*) U computes $\{\{M\}_{SIG_{PN}},r,\{w\}_{ENC_{AI}}\}_{ENC_{AI}}$
(3) $U\Rightarrow AI$: $\{\{M\}_{SIG_{PN}},r,\{w\}_{ENC_{AI}}\}_{ENC_{AI}}$ · · · [그림 1]의 단계 8에 해당
(*) AI computes $z=w^{d_2} \bmod N$
(*) AI records $\langle PN_U:\{z\}_{ENC_{AI}}\rangle$
(4) $AI\Rightarrow U$: z · · · [그림 1]의 단계 9에 해당
(*) U recovers $h^d \bmod N$ by computing $z\cdot r^{-1} \bmod N$

[그림 2] 추적가능 프로토콜

mod$N$를 계산한다. $\langle M, e, N\rangle$를 사용하여 $z \cdot r^{-1}$ mod$N$을 검증하고 $\langle PN_U : \{z\}_{K_{CA}^{-1}}\rangle$를 저장장치에 기록한다. 이것은 향후 분쟁 발생시 추적을 위해서 사용될 값이다. 마지막으로 z를 계산하여 U에게 돌려준다.

(4) $AI \mapsto U : z$

U는 $z \cdot r^{-1}$mod$N$를 계산하여 h^dmod$N$를 밝힌다. $\langle M, e, N\rangle$를 사용하여 h^dmod$N$을 검증하고, 검증에 성공하면 U는 $\langle M, h^d \bmod N\rangle$을 새로운 익명인증서로 사용할 수 있다.

3. 분석

프로토콜 설계의 핵심 아이디어는, 사용자가 인증기관으로부터 발급받은 인증서를 통하여 그 인증 범위에서 새로운 익명 인증서를 발급받되, 익명 인증서 발급자를 익명발급자와 은닉발급자로 나누어 어느 한쪽은 실명만 확인하고 어느 한쪽은 내용만 확인하도록 한 것이다. 즉, 기존의 공개키 기반 구조를 바탕으로 공인인증서를 활용하도록 하며, 익명 인증서 역시 기존의 X.509 인증서 프레임워크를 따르도록 한다. 본 프로토콜이 만족하는 요구 조건에 대해서 살펴보도록 한다.

o 익명 요구 조건

사용자의 익명 인증서(h_i^dmod$N$)와 그 내용 ($M = \langle PN_U, ppk_U, INFO\_ATTR\rangle$)으로부터 사용자에 대한 정보를 유추할 수 없으므로 익명성(Anonymity)을 만족하며, 사용자가 여러 익명 인증서를 사용해도 인증서를 통해, 사용자에 대한 정보를 유추할 수 없으므로 결합익명성(Linkability)도 만족한다.

o 안전성 요구 조건

공개키 인증서에 바탕을 두고 있으므로, 익명 인증서를 이용하여 사용자를 정확히 인증할 수 있다. 따라서 인증성(Authenticity)을 만족한다. 하지만 인증서와 실 사용자를 올바르게 연결하기 위한 방법이 명확하지 않으며, 특히 발급자가 내용에 대한 확인을 전혀 수행할 수 없으므로 책임성(Accountability)을 제공할 수 없다.

또한 같은 이유로 여러 사용자가 결탁하는 경우 사용자의 인증서를 위조할 수 있으므로 위조 방지(Protection against forgery)를 보장할 수 없다.

사용자가 자신의 인증서를 타인과 공유 또는 양도하기 위해서는 반드시 자신의 개인키를 전달해야한다. 이것은 결과적으로 All-or-Nothing의 문제에 해당하므로 기존의 공유 방지[1](Protection against sharing) 요구조건은 만족한다고 할 수 있다.

전자서명의 안전성에 의존하여, 해당 알고리즘이 안전하며 또한 안전하게 구현되었을 경우 인증서에 대한 무단 변조나 변경이 불가능하므로, 인증서의 무결성(Integrity)을 제공한다.

o 추적 및 폐기 요구 조건

비합법적인 사용과 같은 특정 조건 하에서 사용자의 실제 신분을 추적할 수 있어야 하지만, 내용에 대한 확인과 명확한 로그를 남길 수 없으므로 추적성(Traceability)을 제공하지 않는다.

인증서의 비밀키 노출과 같은 특정 조건 하에서 인증서의 기능을 폐기할 수 있어야 하는데, 이것은 공개키 기반 구조에 전적으로 의존하므로, CRL(Certificate Revocation List: 인증서 폐기 목록)을 이용하여 구현할 수 있다. 따라서 폐기성(Revocation)을 제공한다.

o 효율성 요구 조건

전체적인 시스템은 효율적으로 운용될 수 있어야 하고, 중앙 집중에 따른 병목현상을 방지하여 서비스 거부 공격에 내성을 가져야 한다. 일단 발급된 익명 인증서는 향후 서버의 참여를 요구하지 않지만, CRL에 대한 관리는 필요하다.

o 사용회수의 제한

인증서의 이중사용을 방지하여, 한번만 사용하도록 할 수 있기 위한 일회성(One-time show)은 제공할 수 없으며, 내용을 검증할 수 없으므로 각 사용자는 인증서를 오직 한번만 발급받을 수 있도록 하기 위한 유일성(Unicity)도 제공할 수 없다.

o 권한 기반적 접근

발급자를 권한에 따라 분리하고, 각 분리된 발급자는 RSA 다중 서명방식을 사용하도록 하며, 해당 발급자들의 다중 서명을 통해서만 익명 인증서를 얻을 수 있도록

1) 더 높은 요구조건으로서 익명 인증서의 개인키를 공유할 경우, 공인인증서의 개인키가 유도될 수 있도록 구성하는 것을 고려해볼 수 있다. 이것은 All-or-Nothing의 문제를 공인인증서와 연계하여 보다 강력하게 공유 방지를 보장할 수 있도록 하기위한 조건이다. 하지만, 기존의 방법들과 여기서 소개한 기본 프로토콜도 이것을 만족하지는 않는다.

하였다. 발급자 1은 사용자가 제시한 내용은 알지 못하지만 사용자의 신원파악, 즉 인증을 한 후 부분서명을 한다. 발급자 2는 사용자의 신원은 알지 못하지만, 사용자가 제시한 내용 혹은 그 유도값을 검토한 후 부분서명을 한다. (투표와 같이 내용의 비밀보장이 요구될 경우는 유도값만을 제시하도록 하여 그 내용을 숨길 수 있도록 한다.) 안전성 강화를 위하여 비밀 분산의 범위를 넓힐 수 있도록 하며, threshold 기법 등을 통한 기능 강화를 도모할 수 있다.

o 편의성

본 논문에서 제안한 익명 인증서는 X.509 인증서 체계를 따르며 따라서 (PI의 권한이 인정될 경우) 기존 공개키 기반 구조에서 그대로 사용 가능하다는 장점이 있다. 즉, 기존 인증서 사용자는 기존과 마찬가지로 익명 인증서를 편리하게 활용할 수 있다. 앞에서 이미 언급한 바와 같이, 아이디와 패스워드에 익숙한 사용자 역시 해당 서버에 등록할 때만 익명 인증서를 사용하고 이후에는 아이디와 패스워드만으로 편리하게 서비스를 사용하도록 할 수 있다. 이와 같이 기존 시스템 체계에서 편리하게 사용할 수 있다는 것이 본 익명 인증서의 가장 큰 장점이라고 할 수 있다.

o 단점

본 기법의 단점은 익명 인증서, 즉 가명들 간의 비연결성(unlinkability)을 제공할 수 없다는 것이다. 하지만 이것은 기존 X.509 인증서 체계를 유지하는 데에서 비롯된 결과이며, 기존 체계 유지를 고려하는 의미에서 포기할만한 기능이다. 하지만 비연결성을 더욱 중요시하는 응용 분야에서는 X.509 인증서 체계를 포기해야할 것이다.

Ⅳ. 익명게시판 구현

1. 개발 환경

앞에서 소개한 익명인증서 프로토콜을 바탕으로 구현한 익명게시판 사례를 소개한다[10]. 본 익명게시판은 국내 공개키 기반 구조를 고려한 프로토타입으로서 구현되었으며 현재 관련 논문을 작성 중이다. 사용자의 개인키는 국내 블록 암호 표준인 SEED를 이용하여 암호화되며 실명인증서로서 국내의 공인인증서를 사용할 수 있도록 하였다. 익명게시판과 인증서 발급 및 관리 시스템의 구현 환경은 다음과 같다.

* 윈도우즈 XP SP1/2003
* C (Microsoft Visual C++)
* OpenSSL 0.9.7.c
* MySQL 3.23.49
* PHP 4.3.4
* Apache 1.3.29

2. 익명게시판

2.1 구현 모델

익명게시판의 구현 모델은 [그림 3]과 같다. 공통되는 모듈별로 작성하였고, 서버는 Win32 Console Application으로 구현되었으며, 클라이언트는 Internet Explorer와 ActiveX로 구성된다. 로그인서버와 게시판은 하나의 PC안에서 수행되며 DB를 공유하고, 이것들과 AI, BI는 서로 독립된 PC에서 자신만의 DB를 가지고 구동된다.

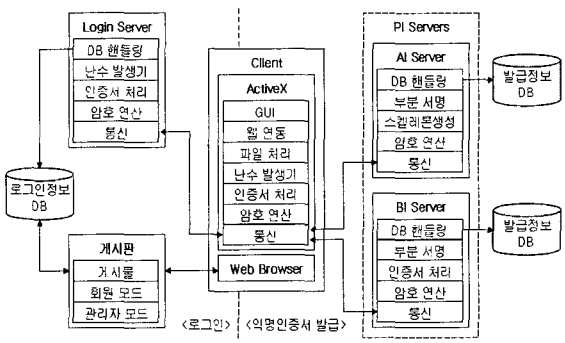
AI와 BI는 III 절에서 소개한 발급절차를 따른다. 로그인 서버는 익명인증서를 검증한 후, 검증에 사용된 값을 DB에 기록한다. 게시판에서는 이 DB를 이용하여 인증된 사용자에게 서비스를 제공한다.

2.2 익명게시판 실행

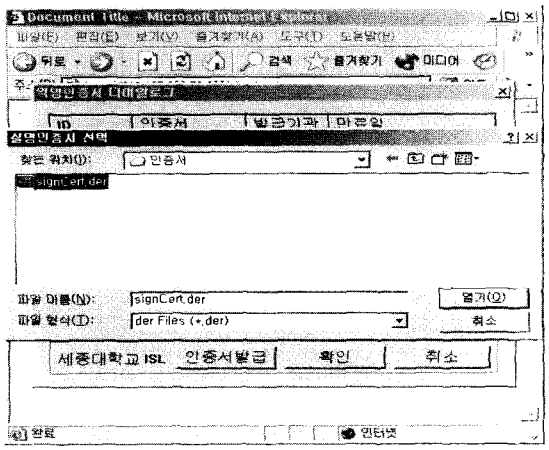
아래 그림은 로그인과 발급 단계에 해당하는 클라이언트와 서버의 수행결과이다. 서버에서 출력되는 값은 헤더파일에서 #define로 결정할 수 있다.

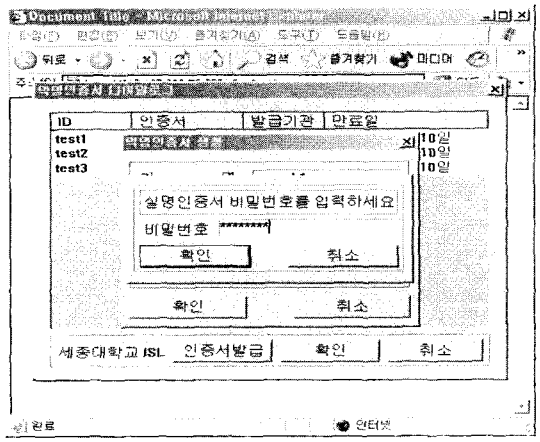
1) 발급 - Client

먼저 클라이언트 창에서 게시판 서버에 접속을 하면 익명인증서 제공을 요구하며, 익명인증서가 없을 경우 새로운 익명인증서를 발급받도록 한다. 이 때 사용자는 아래와 같이 먼저실명인증서를 (국내에서는 공인인증서)



[그림 3] 구현 모델



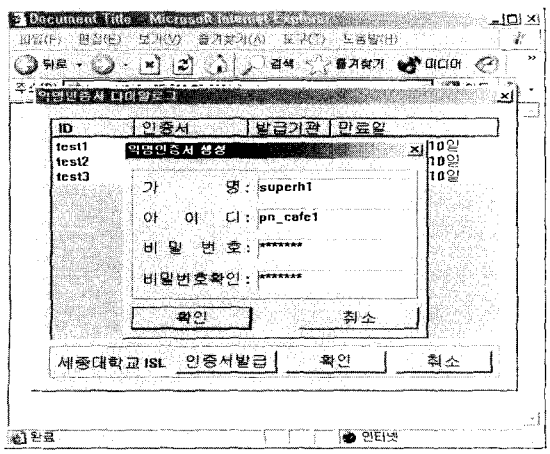


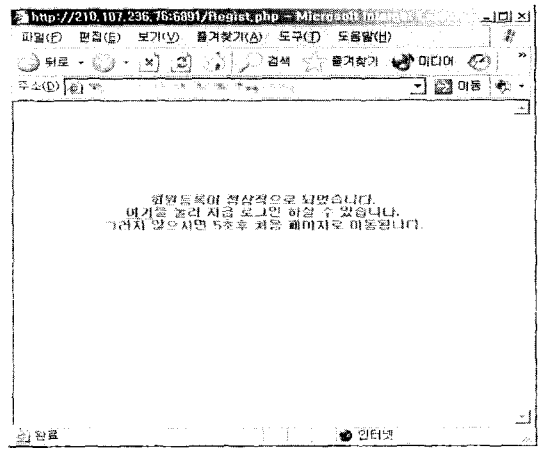
제시해야 한다.

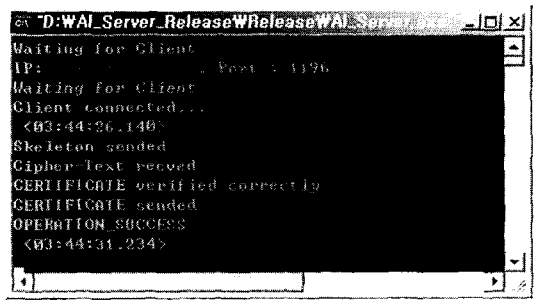
실명인증서를 먼저 제시하는 이유는 새로운 익명인증서 발급 시 BI에게 신원 검증을 받기 위함이다. 이어서 사용자는 아래의 창과 같이 익명인증서에 기입할 자신의 가명을 임의로 선택할 수 있으며, 또한 익명인증서를 관리하거나 아이디-패스워드 인증을 사용할 경우 참조하기 위한 아이디를 임의로 지정할 수 있다. (본 프로토타입에서는 아이디-패스워드 인증을 생략하였다.) 이와 함께 사용자는 익명인증서 개인키를 보관하기 위한 패스워드를 선택할 수 있다.

위의 창에서 확인 버튼을 누르면 클라이언트는 BI와 AI에 차례로 접속하여, 익명인증서를 발급받는다. 이 때 앞에서 언급한 바와 같이 BI에게 신원확인을 받기위하여 실명인증서를 필요로 하게 되며, 따라서 사용자는 아래와 같이 이에 대한 패스워드를 입력해야 한다.

이어서 익명인증서가 무사히 발급되면, 사용자는 다음과 같은 창을 통해서 익명게시판으로 이동할 수 있다.



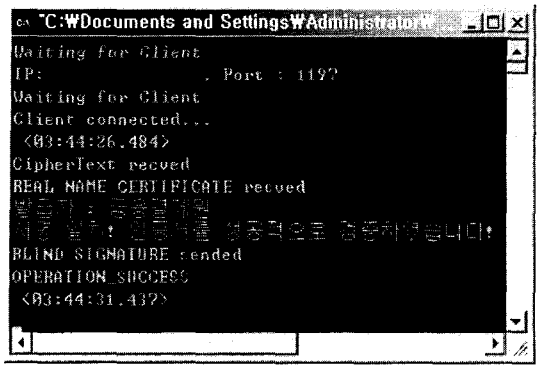


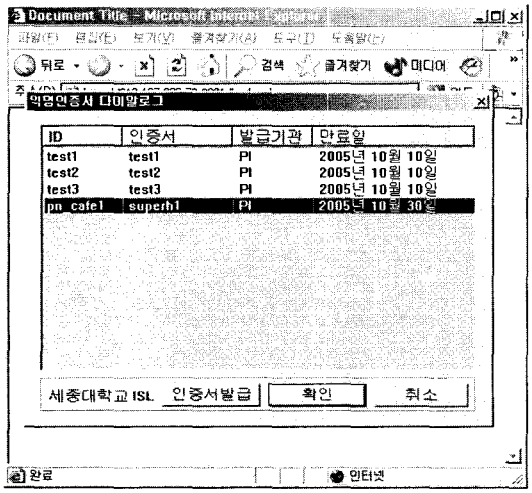


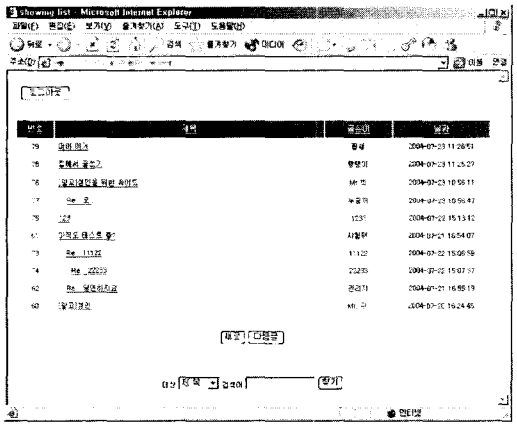
2) 발급 - AI, BI

사용자의 익명인증서 발급시 AI 서버의 실행창에서는 아래와 같은 메시지가 출력된다. 이것은 AI 서버가 정상적으로 수행되었음을 보여주기 위함이다. 포트 1196에 클라이언트가 접속하여 스켈레턴을 발급하였으며, 최종적으로 익명인증서 발급이 완료되었음을 알 수 있다.

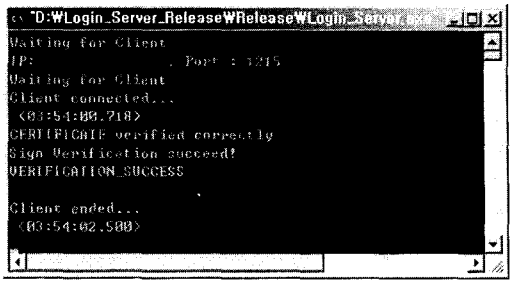
한편 사용자의 익명인증서 발급시 BI 서버의 실행창







에서는 아래와 같은 메시지가 출력된다. 이것은 BI 서버가 정상적으로 수행되었음을 보여주기 위함이다. 포트 1197에 클라이언트가 접속하였으며, 금융결제원 발급 공인인증서 검증이 이루어져, 익명인증서의 부분서명을 제공했음을 알 수 있다.



3) 로그인 - Client

익명인증서를 발급받은 사용자는, 익명게시판 접속시 다음과 같은 창에서 원하는 익명인증서 혹은 원하는 아이디를 선택할 수 있다.

선택한 아이디 혹은 익명인증서에 대한 패스워드를 입력하면 아래와 같은 익명게시판으로 접속할 수 있다.

4) 로그인 - Server

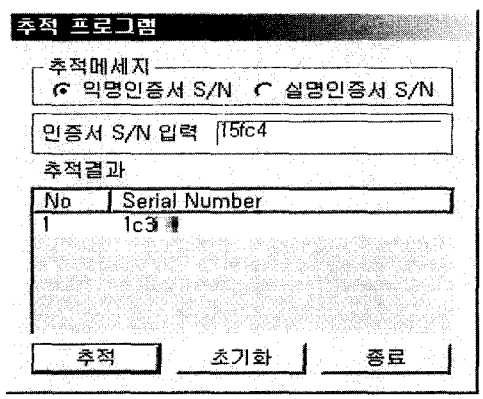
사용자 로그인시 로그인 서버의 실행창에서는 아래와 같은 메시지가 출력된다. 이것은 로그인 서버가 정상적으로 수행되었음을 보여주기 위함이다. 포트 1215에 사용자가 접속하여 인증되었음을 보여준다.

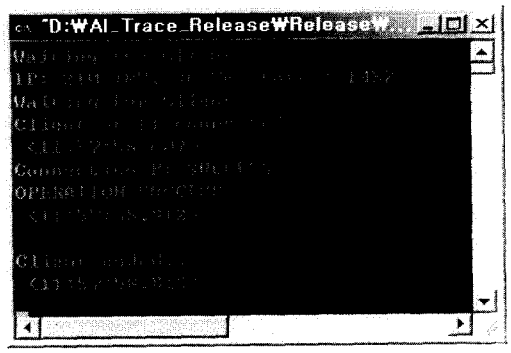
5) 추적 - 익명→실명 - Client

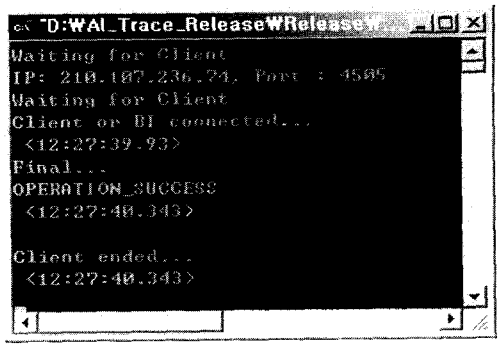
사용자의 실명을 복원하기 위한 추적 프로그램 프로토타입은 다음과 같다. 먼저 실명 복원을 원하는 익명인증서를 선택하여 (인증서에서 판독 가능한) 해당 일련번호를 제공하면 AI와 BI간에 추적 프로토콜이 수행되어, 다음과 같이 해당 실명인증서의 일련번호가 제공된다.

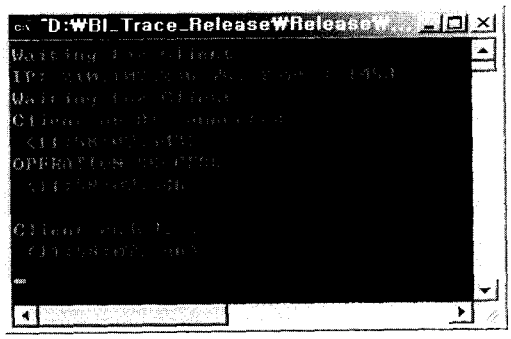
6) 추적 - 익명→실명 - AI BI

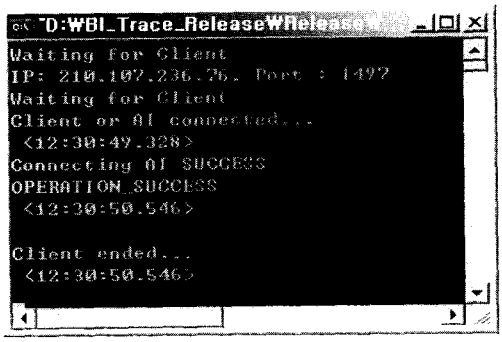
실명 추적시 AI와 BI의 실행창에는 각각 아래와 같은 메시지가 출력된다.









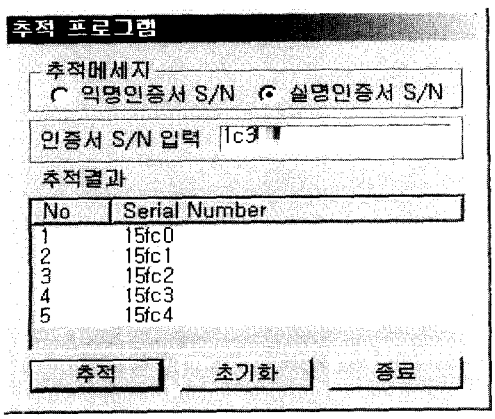


7) 추적 - 실명→익명 - Client

어떤 사용자가 사용하는 모든 익명을 추적하기 위해서는 추적 프로그램 프로토타입을 다음과 같이 실행한다. 즉, 먼저 실명인증서의 (인증서에서 판독 가능한) 해당 일련번호를 제공하면 AI와 BI간에 추적 프로토콜이 수행되어, 다음과 같이 해당 익명인증서의 일련번호들이 제공된다.

8) 추적 - 실명→익명 - AI BI

익명인증서 추적시 AI와 BI의 실행창에는 각각 아래와 같은 메시지가 출력된다.



2.3 성능 측정

성능 측정을 위해서 로그인, AI, BI 서버는 펜티엄4 2.4GHz, 512MB 사양의 PC를 사용하였으며, 운영체제로는 로그인 서버와 AI는 윈도우즈 XP를, BI는 윈도우즈 2003을 사용하였다. 클라이언트는 펜티엄4 1.8GHz, 512MB 사양의 PC에서 윈도우즈 XP를 사용하여 수행되었다. 또한 네트워크는 10M급 중저속장비에 연결하였다.

익명인증서 발급과 로그인시에 AI, BI, 로그인 서버에서 수행된 시간은 다음과 같다. 단위는 ms이다.

클라이언트는 AI와 BI에 먼저 연결 한 후 모든 연산을 시작하고, AI와 BI의 연산은 병렬적으로 이루어진

〔표 1〕 성능 측정

	AI	BI	Login
1회	4547	4375	1809
2회	5047	4801	1766
3회	5516	5360	1770
4회	4984	4828	1828
5회	5094	4953	1782
평균	5,037.60	4,863.40	1,791.00

다. 그러므로 본 프로토타입 시스템에서 익명인증서의 발급을 위해서 클라이언트에서 소요되는 시간은 평균적으로 5초 내외이다. 이것은 프로토타입 시스템에 대한 성능 측정 결과이며, 코드의 최적화와 고속 암호 라이브러리 사용을 통하여 개선할 수 있다. 추적에서 사용되는 연산량은 발급에서 사용되는 연산량보다 적으므로, 별도로 측정하지 않았다.

V. 결 론

인터넷 게시판에서 실명을 사용할 경우 자유로운 토론이 어려우며 사용자 프라이버시를 침해할 우려가 있는 반면, 가명을 사용할 경우 자유로운 토론은 가능하지만 오히려 상호 비방이나 유언비어 등의 부작용이 있을 수 있다. 따라서 기본적으로는 가명을 이용해서 포스팅하도록 허용하지만, 필요한 경우 분산된 여러 개체간의 합의에 의해서 조건부 실명 복원(혹은 다른 말로 조건부 추적)이 가능한 게시판이 요구되고 있다. 또한 최근 전자거래에서 본격적으로 공개키 인증서가 사용되면서 이러한 실명 인증을 통한 개인정보유출이 심각한 문제로 대두되고 있으며 이에 따라 공인인증서의 단점을 보완할 수 있는 익명 인증서의 연구 및 제도 마련이 필요하다. 익명인증서의 필요성은 전자거래뿐 만 아니라 유비쿼터스 컴퓨팅 환경과 같은 미래의 정보 인프라에서는 그 요구가 더욱 증가할 것이다.

본 논문에서는 사용자 익명성을 갖는 기존의 전자서명 인증 기술을 분석하고 이를 통하여 우리 상황에 적합한 익명인증서의 요구조건을 살펴보았다. 또한 이러한 요구조건에 부합되는 RSA에 기반하여 조건부 추적 가능한 새로운 익명인증서 모델과 익명인증서 발급 프로토콜을 살펴보고 분석하였다. 이와 같은 익명인증서 프로토콜은 그 구성이 매우 단순하며 효율적일 뿐만 아니라 기존 X.509 인증서 체계를 유지할 수 있다는 장점이 있다. 즉, 기존 공개키기반구조와 호환성을 제공하고 RSA 전자서명 알고리즘을 지원한다. 제안 기법의 효과적인 응용 분야로 일반 웹사이트 인증, 가명 게시판, 전자 투표, 의약 처방전 등에의 응용이 가능하다. 또한 제안 모델과 프로토콜로부터 몇 가지 확장이 쉽게 가능하다. 첫째, BI의 수를 늘려서 익명성 보장 기능을 강화할 수 있다. 예를 들면, threshold RSA 기법을 통하여 여러 BI (예를 들면, 법원, 공공기관, 시민단체 등등)를 수용할 수 있으며 이를 통해서 더 많은 응용 분야를 개발할 수 있다. 둘째, 익명인증서의 확장필드를 이용하여 다양한 유형의 선택적인 크리덴셜을 구현할 수 있다[9]. 마지막으로 프로토콜을 바탕으로 한 익명인증서 발급 시스템과 익명게시판 구현 사례를 살펴보았다.

참 고 문 헌

[1] S. Brands, "A Technical Overview of Digital Credentials," Preprint, 2002.

[2] J. Camenisch and A. Lysyanskaya, "An Effieient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation," Advances in Cryptology - Eurocrypt 2001, Springer-Verlag, LNCS Vol. 2045, pp. 93-118, 2001.

[3] E. Verheul, "Self-Blindable Credential Certificates from the Weil Pairing," Asiacrypt 2001, Springer-Verleg, LNCS Vol. 2248, pp. 533-551, 2001.

[4] D. Chaum, "Security wihout Identification: Transaction Systems to Make Big Brother Obsolete," Communications of the AGM, Vol. 28, No. 10, pp. 1030-1044, 1985.

[5] D. Chaum and J. Evertse, "A Secure and Privacy-protecting Protocol for Transmitting Personal Information between Organizations," Advances in Cryptology - Crypto'86, Springer-Verlag, LNCS Vol. 740, pp. 118-167, 1987.

[6] I. Damgard, "Efficient Concurrent Zero-knowledge in the Auxiliary String Model," Advances in Cryptology - Eurocrypt 2000, Springer-Verlag, LNCS Vol. 1807, pp. 431-444, 2000.

[7] L. Chen, "Access with Pseudonyms," Cryptogaphy: Polish and Algorithms, Springer-Verlag, LNCS Vol. 1029, pp. 232-243, 1995.

[8] A. Lysyanskyaya, R. Rivest, A. Sahai, and S. Wolf, "Pseudonym Systems," Selected Areas in Cryptography, Springer-Verlag, LNCS Vol. 1758, 1999.

[9] T. Kwon, J. Cheon, Y. Kim and C. Chung, "A Traceable X.509 Pseudonym Certificate for RSA-based PKI," in sub-

mission and available from http://dasan.sejong.ac.kr/~tkwon/research/pseudonym1.pdf, 2003

[10] T. Kwon, J. Cheon, and Y. Kim, "Anonymous Certificate and its Application," in preparation and available from http://dasan.sejong.ac.kr/~tkwon/research/pseudonym2.pdf, 2004

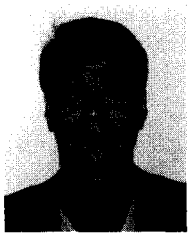
〈著者紹介〉

권 태 경 (Taekyoung Kwon)
종신회원

1988년 3월~1999년 8월 : 연세대학교 컴퓨터과학과 학사, 석사, 공학박사
1999년 9월~2000년 8월 : UC Berkeley 박사후 연구원
2001년 3월~현재 : 세종대학교 컴퓨터공학부 조교수
〈관심분야〉 정보보호, 컴퓨터 네트워크, 암호프로토콜 등

박 해 룡 (Haeryong Park)
종신회원

1999년 2월 : 전남대학교 수학과 학사
2001년 2월 : 서울대학교 수학과 석사
2000년 12월~현재 : 한국정보보호진흥원 연구원
〈관심분야〉 : 암호프로토콜, 키관리, 정보보호



이 철 수 (Chulsoo Lee)
정회원

1975년~1977년 : KAIST 전산과 석사
1977년~1981년 : KAIST 전산과 박사
1982년~1993년 : (주) 데이콤
1993년~1998년 : 한국전산원
1999년~2000년 : 한국정보보호진흥원 원장
2000년~2002년 : 정보통신대학교
2003년~현재 : 경원대학교 소프트웨어 대학 교수
〈관심분야〉 정보보호 정책, 공개키기반구조, 침해사고대응 기술