情報保護學會論文誌
第19卷 第1號, 2009. 2

OT(Oblivious Transfer) 기반의 조건부 추적이 가능한 가명 프로토콜*

강 전 일,[1†] 양 대 헌,[1‡] 이 경 희[2]

[1]인하대학교, [2]수원대학교

Conditionally Traceable Pseudonym Protocol based on Oblivious Transfer*

Jeonil Kang,[1†] DaeHun Nyang,[1‡] KyungHee Lee[2]

[1]INHA University, [2]University of Suwon

요 약

익명성을 지원하는 시스템을 위하여 현재, 익명 신용장 시스템이 많이 연구되고 있다. 그러나 이러한 시스템은 높은 보안 수준을 가질 뿐, 정교한 접근 제어, 필요에 따른 추적 기능 등 실제 응용 환경에서 필요로 하는 특징을 충족시키지 못하는 경우가 대부분이다. 이러한 시스템에 대한 새로운 도전으로써, 복수의 참여자가 가명과 실명에 대한 연결 정보를 분리하여 저장하는 몇몇 연구가 진행되었다. 이 논문에서는 그 중 Oblivious Transfer를 이용한 가명 획득 프로토콜에 기반을 두고, 가명 고갈의 문제를 해결하고, 재암호화(Re-Encyption), 일방향 함수 등을 사용하여 외부에서의 가명의 연결 불가능성과 같은 다른 여러 요구조건을 충족하는 프로토콜을 제안하고 있다.

ABSTRACT

Recently, there have been many researches about anonymous credential systems for supporting the user anonymity. However, these systems only hold a high security level, even though they must be able to be applied to various application that might require access control, conditional traceability, etc. As new challenges to these systems, some researches that several entities store the link information that associates identities and pseudonyms each other have been performed. In this paper, based on the oblivious transfer, we suggest a new pseudonym protocol that solves the pseudonym exhaustion problem which the original pseudonym retrieval protocol suffers from. By using the universal re-encryption and one-way function, we can also archive other requirements like the pseudonym unlinkability from the outside.

Keywords : Anonymous Credential, Pseudonym, Oblivious Transfer

Ⅰ. 서 론

개인 정보보호와 관련한 익명(Anonymity)과 가명(Pseudonym)에 관한 연구는 1980년대 후반으로 거슬러 올라간다. Chaum은 1985년 가명 시스템(Pseudonym

접수일(2008년 9월 1일), 게재확정일(2008년 12월 17일)
* 본 연구는 정보통신부 및 정보통신연구진흥원의 IT신성장동력핵심기술개발사업의일환으로 수행하였음.
(2008-F- 036-01, 익명성기반의 U-지식 정보보호기술개발)
† 주저자, dreamx@seclab.inha.ac.kr
‡ 교신저자, nyang@inha.ac.kr

System)을 소개하였다[1]. 가명 시스템 또는 익명 신용장 시스템(Anonymous Credential System)에서 사용자는 기관에 따라 여러 개의 가명(Pseudonym, 또는 Nym)을 소유할 수 있고 기관은 사용자를 가명으로 인식한다. 기관은 가명을 가진 사용자에게 신용장(Credential)을 발급해주며, 사용자는 가명의 노출 없이 신용장의 소유를 다른 기관에게 증명함으로써, 자신이 최초에 성공적으로 인증을 수행했음을 보인다. 이후의 많은 연구에서 가명 시스템이 실제로 구현되었으며, 여러 문제점이 수정되어 왔다[2,3,4,5,6]. 그러나 이러한 익명 신용장 시스템은 지나치게 높은 보안 수준을 충족하도록 설계된 나머지, 정교한 접근 제어, 필요에 따른 추적 기능 등 실제 응용 환경에서 필요로 하는 특징을 충족시키지 못하는 경우가 대부분이었다. 몇몇 시스템은 보안성이나 연산을 희생하는 대신 이러한 문제를 해결하려는 노력이 있었다[7,8].

한 편, 최근에는 수학적 어려움을 이용하여 설계된 가명과 신용장 등을 이용하여 익명성을 확보하는 기존의 익명 신용장 시스템과는 다르게 구조적인 특징을 이용하여 익명성을 확보하는 익명 시스템이 설계되고 있다. 이러한 새로운 익명 시스템은, 익명 신용장 시스템처럼 가명 간의 연결성조차 허용하지 않는 수준의 보안성을 가지고 있지 않지만, PKI(Public Key Infrastructure) 등과 결합이 쉽고, 실제 응용 환경에서 사용될 수 있는 특징들을 갖고 있는 장점이 있다. Benjumea 등은 공정한 블라인드 서명(Fair Blind Signature)과 공개 가명과 비공개 가명, 실명과 가명의 연결 정보의 분할을 통하여 속성 인증서(Attribute Certificate)을 공개키 인증서(Public Key Certificate) 없이 사용하는 방법에 대해서 제안하였다[9,10]. 권태경 등은 PKI에서 인증서의 발행 주체를 둘로 나누고, 실명을 두개로 나뉜 비밀키로 암호화하여 이를 두 주체가 각각 나누어 갖도록 하여 익명성을 확보하는 프로토콜을 소개하였다[11]. 양대헌 등은 OT (Oblivious Transfer)[12,13]를 이용하여 서비스 제공자로부터 가명을 익명으로 획득하고, 이를 제한적으로 추적할 수 있는 가명 획득 프로토콜을 소개하였다[14]. 이러한 기법과 프로토콜들의 공통점은 실명과 가명의 연결 정보(또는 실명의 복호화 정보)가 둘 이상의 참여자에게 분할하여 저장함으로써 사용자에 대한 익명성을 확보하고, 데이터베이스의 통합을 통하여 실명과 가명을 연결시켜 익명성을 철폐하는 메커니즘을 가지고 있다는 것이다.

OT를 이용한 가명 획득 프로토콜은 적극적으로 가명을 획득하고 사용하는 방법에 대해서 기술하고 있다. 서비스 제공자는 기존의 익명 기법들처럼 CA(Certificate Authority)와 같은 특별한 참여 개체를 전적으로 신뢰하지 않아도 되는 장점을 가지고 있고, 단지 가명의 추적을 위하여 판단기관(Judicial Authority)을 두고 있을 뿐이다. 그러나 이 프로토콜은 다른 익명 시스템에서와 다르게 가명 풀(Pseudonym Pool)을 유지해야 한다. 만약 사용자가 지나치게 많아서 가명 풀이 모두 고갈되었을 때에는, 가명을 이용하여 사용자를 구별할 수 없게 된다. 이는 실명 추적이 더 이상 유효하지 않음을 의미한다.

이 논문에서는 일반적 재암호화(Universal Re-Encryption)[15]를 응용하여 가명 풀의 고갈 문제점을 해결하는 방법을 제시한다. 사용자는 OT를 이용하여 복수의 예비 가명을 서비스 제공자로부터 얻은 뒤, 이를 취합하여 가명을 얻는다. 이 과정에서 발생할 수 있는 사용자간의 공모를 통한 유효한 가명의 생성을 하지 못하도록 일반적 재암호화를 응용한다. 2장에서는 이 논문에서 제안하는 프로토콜이 이루고자 하는 바와 프로토콜을 설명하기 위해 알아두어야 할 기호 등을 설명한다. 3장에서는 OT와 일반적 재암호화를 이용한 가명 프로토콜을 기술한다. 4장에서는 3장에서 기술한 프로토콜이 갖는 특징과 보안성에 대해서 검토한다. 5장에서 결론으로 논문을 마무리한다.

Ⅱ. 프로토콜의 개요

2.1 제안 프로토콜의 목적

OT를 이용한 가명 획득 프로토콜[14]에서는 판단기관(Judicial Authority)과 서비스 제공자(Service Provider), 사용자(User)가 참여한다. 사용자는 우선 판단기관과 서비스 제공자에 실명으로 인증을 수행하고 보안 세션을 만들어둔다. 이후의 통신은 이 보안 세션을 통하여 이루어진다. 실명 인증이 끝나면 판단기관는 서비스 제공자가 가진 가명들의 색인(Index)을 m개 선택하여 사용자에게 전송해준다. 사용자는 이 m개의 색인에 해당하는 가명과 가명의 인증키 쌍을 서비스 제공자로부터 OT를 통해 전송 받는다. 그 후 사용자는 판단기관에게 전송받았던 색

인과 임시로 암호화된 가명의 인증키를 다시 돌려보낸다. 판단기관은 이 중에서 1개를 제외한 나머지의 암호화 비밀키를 사용자에게 요구하고 이를 얻어 가명의 인증키들을 복호화 한다. 판단기관은 이 값들을 색인과 함께 서비스 제공자에게 전송하고, 서비스 제공자는 가명의 인증키가 올바른지 확인한 후 결과를 판단기관에게 알려준다. 만약 모두 올바르다면 판단기관은 사용자에게 OK 메시지를 보내고 사용자는 남은 하나의 가명을 사용한다.

이와 같은 프로토콜에서 판단기관과 서비스 제공자는 각각 색인 풀(Index Pool)과 가명 풀을 관리해야한다. 그러나 풀의 크기는 고정되어 있으므로, 사용자가 매우 많을 때 풀이 고갈되는 문제가 발생할 수 있다. 이를 방지하기 위해서는 풀이 고갈되기 전에 풀을 늘려주는 등의 추가적인 조치가 필요하다. 그러나 이러한 경우, 풀이 늘어나기 전과 후에 사용되는 가명이 확연히 달라지기 때문에 사용자의 익명성이 낮아지게 된다.

따라서 이 논문에서 이루고자 하는 목표는 다음과 같다.

- **예비 가명의 조합을 통한 가명 고갈 문제의 해소:** 가명을 조합하여 사용함으로써, 기존 프로토콜과 비교하여 더 많은 사용자에게 가명을 할당하는 것이 가능하다.
- **사용자의 공모 방지:** 사용자들끼리 공모하여 유효한 제 3의 가명을 만들어내지 못한다.
- **가명의 위조 방지:** 사용자는 서비스 제공자로부터 할당받은 가명을 변경하거나 유효한 가명으로부터 다른 유효한 가명을 만들어내지 못한다. 이는 사용자가 익명성을 악용하여 일종의 Sybil 공격을 수행하지 못하도록 한다.
- **외부에서의 가명 연결 불가능성:** 사용자가 가명을 사용할 때, 외부에서는 사용자의 가명을 서로 연결하지 못한다.
- **내부에서의 가명 연결 가능성:** 서비스 제공자는 사용자의 가명을 연결할 수 있다.
- **제한적 실명 추적 가능성:** 서비스 제공자는 판단기관과 협력을 통하여 사용자의 익명성을 철폐할 수 있다.
- **가명 철폐 가능성:** 서비스 제공자는 스스로의 판단으로 특정 가명을 철폐할 수 있다.

2.2 프로토콜의 전체 개요

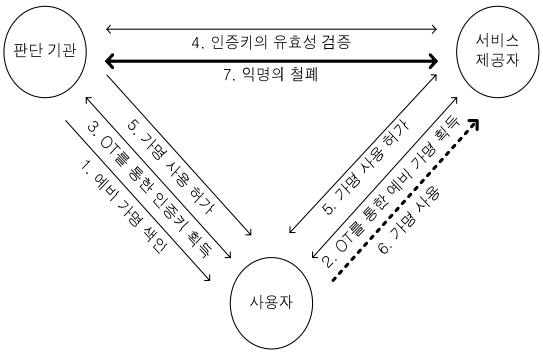
프로토콜의 전체적인 모습은 기존과 대체로 동일하나, 이 논문에서 제안하는 프로토콜에서는 사용자가 판단기관과 서비스 제공자 모르게 가지고 있는 예비 가명의 수가 시스템 파라미터에 따라 일정 수로 늘어나게 된다. 또한 가명의 획득에서만 그치지 않고 사용자가 가명을 사용하는 방법이 기술된다. 추가적으로 가명을 사용하기 위해서는 판단기관이 서비스 제공자에게 이를 등록해야만 한다.

[그림 1]은 이 논문에서 제안하는 가명 획득 프로토콜의 개요 및 동작 순서를 나타낸다. 위 그림에서 1부터 5단계까지는 가명의 획득 단계로, 모두 안전한 통신 채널을 이용하여 이루어진다. 따라서 외부에서 이를 엿듣는 것은 불가능하다. 6단계는 가명의 사용 단계로, 1부터 5단계까지와는 다르게 시간이 흐른 뒤에 이루어지고 안전한 통신 채널을 사용하지 않아도 된다.

2.3 재암호화와 Oblivious Transfer

재암호화[14]는 비밀을 모르는 참여자가 비밀키를 알고 있는 사용자가 만든 암호문을 해당 비밀키를 모르는 상태로 새로운 형태의 암호문을 만드는 기법이다. 어떤 난수 r를 선택하여 비밀키 x와 함께 세션 비밀키 g^{rx}를 만들고, g^r을 첨부함으로써 비밀키를 모른다고 하더라도 새로운 난수를 세션 비밀키에 첨부하는 방법으로 재암호화가 가능하다.

OT(Oblivious Transfer)[12,13]는 송신자가 일정 수의 정보 중에 일부분을 수신자에게 보내주는 암호학적 프로토콜이다. OT가 다른 전송 프로토콜과 특별히 다른



[그림 1] 제안 가명 획득 및 사용 프로토콜의 개요 및 동작 순서

점은, 송신자는 수신자가 일정수의 비밀을 가져갔다는 사실을 알 수 있지만 그것이 무엇인지 알 수 없다는 것이다.

2.4 프로토콜 기술에 필요한 기호 등

이 논문에서 제안하는 프로토콜을 기술하기 위하여 [표 1]과 같은 기호를 사용한다. 프로토콜에서 사용하는 변수(즉, 비밀키, 난수 등)는 본문에서 설명한다.

Ⅲ. 제안 프로토콜

3.1. 준비 과정

위수(order)가 q인 곱셈 순환 그룹(Multiplicative Cyclic Group) $\mathbb{G}$를 가정하자. g는 그룹 $\mathbb{G}$의 생성자이다. 서비스 제공자는 자신의 비밀키 $x \in_R \mathbb{Z}_q$와 $b \in_R \mathbb{Z}_q$를 선택하고, 예비 가명 풀 $L_{v:(c,e,s)} = \{v_1 : (c_1, e_1, s_1), \cdots,$ $v_{\max} : (c_{\max}, e_{\max}, s_{\max})\}$을 준비한다. 이때, $a_i \in_R \mathbb{Z}_q$에 대하여 $c_i = g^b g^{a_i x}$와 $e_i = g^{a_i}$를 만족하며, $s_i \in_R \mathbb{Z}_q$이다. 서비스 제공자 SP는 색인 풀 $L_v = \{v_1, \cdots, v_{\max}\}$과 부분 예비 가명 풀 $L_e = \{e_1, \cdots, e_{\max}\}$를 판단기관과 공유한다. 초기에는 $1 \le i \le \max$에 대해서 $v_i = i$일 수 있으나(즉, $v_1 = 1, v_2 = 2, \cdots$ 등), 이후에 풀에서 예비 가명이 제거되었을 때, 이 값이 바뀔 수도 있다. 판단기관은 서비스 제공자와 공유한 L_v를 색인 풀로 사용한다. 사용자는 서비스 제공자와 판단기관에 각각 실명으로 인증을 시도하고, 안전한 통신 채널을 확보한다. 또한 판단기관과 서비스 제공자는 자신들의 서명을 위한 비밀키를 가지고 있고, 서로의 공개키를 알고 있다.

[표 1] 기술 기호와 그에 대한 설명

기호		설명
함수	$(b_1, b_2, \cdots, b_m) = OT_n^m(a_1, a_2, \cdots, a_n)$	m-out-of-n Oblivious Transfer. $\{a_i\}$는 선택할 수 있는 비밀, $\{b_i\}$는 선택되어 전송된 값
	$H(a)$	$H: \mathbb{G} \rightarrow \mathbb{G}$인 일 방향 함수
	$\Sigma(a,x)$	a의 비밀키 x에 의한 재암호화
	$\exp(a,k)$	a의 k 제곱
	$n(A)$	집합 A의 원소의 개수
	$Cert_X(a)$	a와 참가자 X에 의한 a의 서명 값
	$Log(X)$	가명 X를 사용하는 사용자의 로그
레코드 구조 정의 및 선택	$\alpha_i : \beta_i$	요소 β_i는 인덱스 α_i에 연결되어 있음
	L_α	요소 α에 대한 집합
	$L_\alpha \Leftarrow L_\beta$	요소 β에 해당하는 요소 α의 집합
식별자	JA	판단기관의 식별자
	SP	서비스 제공자의 식별자
	U	사용자의 식별자
시스템 파라미터	max	최대 예비 가명 풀의 크기
	m, n	시스템 파라미터, $m \ll n \ll \max$
메시지	OK	가명 획득 절차의 확인 메시지
	REQ	가명 획득 절차의 시작 메시지
기타	$\rightarrow$, $\leftarrow$	메시지의 전송 방향
	$time$	현재 시간의 타임스탬프

3.2 제안 프로토콜의 동작 순서

3.2.1 가명의 획득

1-1) $U \rightarrow JA$: REQ, SP

1-2) $JA \rightarrow U$: $Cert_{JA}(L_v', U, time)$, L_v''

판단기관은 L_v로부터 무작위로 $L_v' \subset L_v$와 $L_v'' \subset L_v'$을 선택한 후 이를 사용자에게 전송한다. 이 때, $m = n(L_v'') \ll n = n(L_v') \ll \max = n(L_v)$인 관계를 만족한다.

2-1) $U \rightarrow SP$: $Cert_{JA}(L_v', U, time)$

사용자는 판단기관으로부터 받은 $Cert_{JA}(L_v', U, time)$을 서비스 제공자에게 전송한다.

2-2) SP

서비스 제공자는 $Cert_{JA}(L_v', U, time)$이 유효한지 확인한다. 유효하지 않다면 이하의 프로토콜을 수행하지 않는다. 만약 유효하다면 서비스 제공자는 예비 가명 풀 $L_{v:(c,e,s)}$로부터 L_v'에 해당하는 $L_{v:(c,s)}' = \{(v_1' : (c_1', s_1')), \cdots, (v_n' : (c_n', s_n'))\}$을 선택하고, 무작위로 선택한 $k \in_R \mathbb{Z}_q$에 대하여 $L_{v:\exp(c,k)}' = \{(v_1' : (c_1')^k), \cdots, (v_m' : (c_n')^k)\}$를 생성한다. 여기서 $\exp(c_i, k) = (c_i)^k = g^{bk} g^{akx}$와 같다.

2-3) $SP \rightarrow U$: $\Sigma(k,x)$

서비스 제공자는 $(r_0, r_1) \in_R \mathbb{Z}_q^2$를 선택하고 이를 이용하여 $\Sigma(k,x) = \{(\alpha_0, \beta_0) = (kg^{r_0 x}, g^{r_0}), (\alpha_1, \beta_1) = (g^{r_1 x}, g^{r_1})\}$처럼 k를 사용자에게 서명과 함께 전송

한다. 사용자는 가명을 사용할 때 이용하기 위하여 $\Sigma(k,x)$를 반드시 저장해야한다.

2-4) $U \leftarrow SP$: $(L_{v:(\exp(c,k),s)}'' \Leftarrow L_v'')$
$= OT_n^m(L_{v:(\exp(c,k),s)}' \Leftarrow L_v')$
사용자와 서비스 제공자는 $L_{v:(\exp(c,k),s)}' = \{L_{v:\exp(c,k)}', L_{v:s}'\}$에 대한 OT를 수행한다. 사용자는 L_v'으로 지정된 $L_{v:(\exp(c,k),s)}'$로부터 L_v''에 따라 $L_{v:(\exp(c,k),s)}'' = \{L_{v:\exp(c,k)}'', L_{v:s}''\}$를 선택하여 가져온다.

2-5) U
사용자는 $L_{v:(\exp(c,k),s)}'' = \{L_{v:\exp(c,k)}'', L_{v:s}''\}$의 $L_{v:\exp(c,k)}''$로부터 $C^k = (c_1'')^k \times \cdots \times (c_m'')^k$를 계산한다.

3) $U \rightarrow JA$: $L_{v:s}''$ 사용자는 판단기관에게 L_s''를 전송한다.

4-1) $JA \rightarrow SP$: L_v'
판단기관은 서비스 제공자에게 L_v'를 보낸다.

4-2) $JA \leftarrow SP$: $L_{v:s}'' = OT_n^m(L_{v:s}' \Leftarrow L_v')$
판단기관과 서비스 제공자는 $L_{v:s}'$에 대한 OT를 수행한다. 판단기관은 $L_{v:s}'$중에서 L_v''에 따라 $L_{v:s}''$를 선택하여 가져온다.

4-3) JA
판단기관은 사용자로부터 전송받은 $L_{v:s}''$과 서비스 제공자로부터 OT를 통해 얻은 $L_{v:s}''$를 서로 비교하여 사용자가 올바른 예비 가명을 서비스 제공자로부터 얻어왔는지 확인한다.

5-1) $JA \rightarrow U$: $Cert_{JA}(OK, U, time)$, $H(E)$ (또는 $JA \rightarrow SP$: U)
사용자가 올바른 예비 가명을 가져왔다면 판단기관은 사용자에게 $E = e_1'' \times \cdots \times e_m''$를 생성하고 사용자에게 $Cert_{JA}(OK, U, time)$와 $H(E)$를 전송한다. $Cert_{JA}(OK, U, time)$에는 시간, 사용자의 식별자 등의 정보 이외에 필요에 따라 다른 메시지를 담고 있을 수 있다. 판단기관은 사용자의 실명과 L_v''를 이용해서 구한 $E = e_1'' \times \cdots \times e_m''$를 연관시켜 저장한다. 사용자가 올바른 예비 가명을 가져오지 못했다고 판단한 경우 판단기관은 서비스 제공자에게 현재 프로토콜을 수행하고 있는 사용자의 식별자 U를 전송함으로써, 이후의 사용자와의 통신을 중지시키도록 요청한다.

5-2) $U \rightarrow SP$: $Cert_{JA}(\mathrm{OK}, U, time)$
사용자는 서비스 제공자에게 판단기관으로부터 받은 $Cert_{JA}(\mathrm{OK}, U, time)$를 재전송한다.

5-3) $SP \rightarrow U$: $L_{v:\exp(e,k)}' \Leftarrow L_v'$
서비스 제공자는 사용자로부터 받은 $Cert_{JA}(\mathrm{OK}, U, time)$가 올바른지 확인하고, 올바른 경우에 한하여 L_v'에 의해서 지정된 $L_{v:e,}' = \{(v_1' : e_1'), \cdots, (v_n' : e_n')\}$에 대해서 $L_{v:\exp(e,k)}' = \{(v_1' : (e_1')^k), \cdots, (v_n' : (e_n')^k)\}$를 계산하여 사용자에게 전송한다. 여기서 $\exp(e_i, k) = (e_i)^k = g^{ak}$와 같다.

5-4) U
서비스 제공자로부터 $L_{v:\exp(e,k)}'$를 받은 사용자는 $(L_{v:\exp(e,k)}'' \Leftarrow L_v'') \subset L_{v:\exp(e,k)}'$를 구하고 $E^k = (e_1'')^k \times \cdots \times (e_m'')^k$를 계산한다.

3.2.2 가명의 사용

일정 시간이 흐른 후, 사용자는 서비스 제공자에게 가명을 사용하여 서비스를 요청한다. 가명을 획득 후 바로 서비스를 요청할 경우 프로토콜 외적인 부분에서 사용자의 실명이 노출될 가능성이 매우 높다.

6-1) $U \rightarrow SP$: C', E', (α_0', β_0'), (α_2, β_2)
사용자는 획득한 가명을 사용하기 위하여 $r, r' \in_R \mathbb{Z}_q$을 선택하고 $C' = (C^k)^r$, $E' = (E^k)^r$, $(\alpha_0', \beta_0') = (r\alpha_0\alpha_1^r, \beta_0\beta_1^r)$, $(\alpha_2, \beta_2) = (H(E)\alpha_1^{r'}, \beta_1^{r'})$를 계산하여 SP에게 전송한다.

6-2) SP
사용자가 보낸 메시지로부터 $w = \alpha_0'/(\beta_0')^x$, $h = \alpha_2/(\beta_2)^x$를 계산하고 $(g^{bm})^w = C'/(E')^x$인지 확인하고 그렇지 않다면 프로토콜을 중지한다. 모두 옳다면 가명으로 $P = (E')^{1/w}$로 계산하고 $H(P) = h$인지 확인한다. 서로 값이 다르다면 역시 프로토콜을 중지한다. 가명 P에 대한 레코드에 E' 또는 β_0'이 기록되어 있다면 이를 재전송 공격으로 간주하고 이후의 통신을 차단한다. 그렇지 않다면 해당 레코드에 E'와 β_0'을 새롭게 기록하고 이후의 접속을 허용한다. 이후의 행동은 로그로써 남겨둔다.

3.2.3 사용자 실명의 조건부 추적

사용자의 실명을 추적하기 위해서는 판단기관과 서비스 제공자의 협력이 필요하다. 서비스 제공자는 가명 P에 대한 접속 로그 $Log(P)$를 가지고 있다. 이를 근거로 판단기관에게 협력을 요청하면 판단기관은 이를 확인하고 서비스 제공자에게 실명 정보를 제공한다. 판단기관은 스스로의 $Log(P)$ 분석 결과, 서비스 제공자의 요청을 거부할 수 있다.

7-1) $SP \rightarrow JA$: P, $Log(P)$
서비스 제공자는 사용자의 가명 P와 해당 사용자의 $Log(P)$를 판단기관에게 전송한다.

7-2) $JA \rightarrow SP$: U
판단기관은 $Log(P)$를 분석하고 P와 일치하는 E를 가진 사용자의 식별자 U를 서비스 제공자에게 전송한다.

Ⅳ. 제안 프로토콜의 검토 및 토의

4.1 보완 설명 및 수식의 검증

사용자가 익명으로 사용하는 C와 E^k는 다음과 같은 형태를 가지고 있다.

$$C^k = (c_1'')^k \times (c_2'')^k \times \cdots \times (c_m'')^k = g^{bkm} g^{(a_1'' + a_2'' + \cdots + a_m'')xk} \quad (1)$$

$$E^k = (e_1'')^k \times (e_2'')^k \times \cdots \times (e_m'')^k = g^{(a_1'' + a_2'' + \cdots + a_m'')k} \quad (2)$$

(α_0', β_0')은 사용자가 선택한 난수 r과 SP가 선택한 난수 k의 곱에 대하여 SP의 비밀키 x를 이용해 재암호화된 메시지와 형태가 같다. 사용자는 SP로부터 받은 $UE(k,x) = \{(\alpha_0, \beta_0), (\alpha_1, \beta_1)\}$를 이용하여 이러한 메시지를 만들 수 있다. 그러나 보통의 재암호화와 다르게 사용자는 r을 메시지의 앞에 한 번 더 곱해준다.

$$(\alpha_0', \beta_0') = (r\alpha_0\alpha_1^r, \beta_0\beta_1^r) = \left(rkg^{r_0 x}(g^{r_1})^r, g^{r_0}(g^{r_1})^r\right) \quad (3)$$

사용자가 SP에게 C', E', (α_0', β_0')를 전송한 뒤 SP는 다음과 같이 자신의 비밀키 x를 이용하여 $w = \alpha_0' / (\beta_0')^x$와 $W = C' / (E')^x$를 구한다.

$$W = \frac{C'}{(E')^x} = \frac{\left(g^{bkm} g^{(a_1'' + \cdots + a_m'')xk}\right)^r}{\left(g^{(a_1'' + \cdots + a_m'')k}\right)^{rx}} = g^{bkmr} \quad (4)$$

$$w = \frac{\alpha_0'}{(\beta_0')^x} = \frac{rkg^{r_0 x}(g^{r_1 x})^r}{(g^{r_0}(g^{r_1})^r)^x} = rk \quad (5)$$

SP는 자신의 비밀키 b에 대해서 이미 g^{bm}을 알고 있으므로 사용자가 올바른 C와 E^k와 $UE(k,x)$를 사용한다면

$$(g^{bm})^w = W \quad (6)$$

인 관계가 항상 만족 한다. 최종 가명은 $P = (E')^{1/w}$로 결정되는데, 이 값이 JA가 정한 가명과 일치하는지 확인하기 위한 절차가 필요하다. (α_2, β_2)는 가명의 해시 정보가 담겨져 있으며 이를 비밀키 x로 복호화하면 가명 E의 해시 정보를 얻을 수 있다.

$$h = \frac{\alpha_2}{(\beta_2)^x} = \frac{H(E)\alpha_1^r}{(\beta_1^r)^x} = \frac{H(E)g^{x r_1 r}}{(g^{r_1 r})^x} = H(E) \quad (7)$$

최종적으로 얻은 가명이 JA가 정한 가명과 일치한다면

$$h = H(P) \quad (8)$$

와 같아야 한다.

4.2 익명성

- **사용자의 가명의 결정:** 이 프로토콜에서 사용자의 가명은 판단기관에 의해서 결정된다. 가명은 판단기관이 선택한 L_v''에 의해서 선택되고 L_v과 L_v', L_v''의 관계에 의해서 서비스 제공자로부터의 사용자 익명성이 만들어지게 된다. 만약 판단기관이 선택한 L_v'와 L_v''가 사용자마다 배타적으로 선택될 경우 서비스 제공자는 사용자의 실명을 추측할 수 있게 된다.
- **OT를 통해 얻을 수 있는 익명성 정도:** 서비스 제공자의 입장에서 바라보았을 때, 사용자가 가져가는 예비 가명 (L_e''와 L_e'')과 판단기관이 사용자의 예비 가

명의 유효성을 검증하기 위하여 가져간 L_s''는 동일한 OT를 두 번 수행한 것이다(추출을 시도하는 대상(즉, L_v')과 추출되는 대상(즉, L_v'')이 동일하다). 그러나 서비스 제공자는 두 번의 동일한 OT가 발생하였다 하더라도, 서비스 제공자는 사용자가 어떠한 예비 가명을 가져갔는지 확신하기 어렵다. 따라서 서비스 제공자가 바라보았을 때 사용자는 최대 n^m개의 가명을 가질 수 있다.[1] 이는 최대 ${}_nC_m$개의 가명을 가질 수 있었던 기존 OT 기반 가명 획득 프로토콜[14]에 비해서 매우 큰 수준이다.

- **가명의 사용에 따른 익명성:** 외부에서 바라보았을 때, 사용자의 가명을 알아내거나, 가명끼리 연결하는 것은 매우 힘들다. 가명은 매번 다른 난수로 재암호화되어 서비스 제공자에게 전송되므로 옆에서 메시지를 엿듣는 공격자가 있다하더라도 서로 이를 연결하기 위하여서는 사용자가 선택하는 난수 r을 알아내야 하지만 이는 DLP(Discrete Logarithm Problem)를 푸는 것과 같다. 여기서 주의해야할 것으로 익명성의 최대 수혜자는 사용자이므로 사용자가 동일한 난수를 선택하여 가명의 익명성이 취소되는 것의 책임은 어디까지나 사용자에게 있다는 것이다. 서비스 제공자의 입장에서, 서비스 제공자는 사용자의 실명과 연결된 k을 알고 있으므로 사용자의 가명이 어떠한 k에 의해서 변형되어 있는지 확인하면 서비스 제공자는 사용자의 실명을 알아낼 수 있다. 그러나 서비스 제공자는 이를 확인할 수단이 없으므로 가명과 실명을 연결시킬 수 없다(사용자가 전송한 가명 정보에서 k는 언제나 사용자가 선택한 r과 연결되어 있기 때문이다).

4.3 가짜 가명 생성 방지

기존의 OT 기반 가명 획득 프로토콜은 사용자가 α/m의 확률로 판단기관과 서비스제공자를 속이고 다른 가명을 사용함으로써 완벽한 익명성을 손에 넣을 수 있었다. 그러나 제안하는 프로토콜에서는 이것이 불가능하다.

- **사용자간 공모 방지**: 식 (1), (2)와 같이 사용자가 갖게 되는 가명과 가명의 유효성 정보는 서비스 제공자가 생성하는 난수 k에 따라 달라진다. 이 때문에 공모자들은 자신이 알고 있는 예비 가명을 서로 합칠 수 없다. 공모자 A와 B가 얻은 예비 가명을 각각 $\{c_i^{k_A}\}$와 $\{c_i^{k_B}\}$라고 할 때, 이들이 만들 수 있는 가명은 $C_{AB}=g^{bk_A(m-j)}g^{xk_A(\cdots)}g^{bk_Bj}g^{xk_B(\cdots)}$와 $E_{AB}=g^{k_A(\cdots)}g^{k_B(\cdots)}$같은 형태가 된다. 따라서 이를 전송 받은 서비스 제공자는

$$W_{AB}=(C_{AB})^r/((E_{AB})^r)^x = \frac{g^{brk_A(m-j)}g^{xrk_A(\cdots)}g^{brk_Bj}g^{xrk_B(\cdots)}}{\left(g^{rk_A(\cdots)}g^{rk_B(\cdots)}\right)^x} = g^{br(k_A(m-j)+k_Bj)} \quad (9)$$

을 계산하게 된다. 식 (6)이 만족해야 하므로 $g^{bmw}=g^{br(k_A(m-j)+k_Bj)}$이어야 하며, 이는 곧, $mw=r(k_A(m-j)+k_Bj)$와 같음 의미한다. 단순하게 $j=m/2$, $r=2$로 놓는다면 $w=k_A+k_B$와 같아야 한다. 공모자들은 $\Sigma(k_A,x)=\left\{(k_Ag^{r_{A0}x},g^{r_{A0}}),(g^{r_{A1}x},g^{r_{A1}})\right\}$와 $\Sigma(k_B,x)=\left\{(k_Bg^{r_{B0}x},g^{r_{B0}}),(g^{r_{B1}x},g^{r_{B1}})\right\}$를 알고 있다. 이 중 k_A와 k_B에 연관된 내용은 $k_Ag^{r_{A0}x}$와 $k_Bg^{r_{B0}x}$인데 $r_{A,0}$와 $r_{B,0}$을 알 수 없기 때문에, 이로부터 공모자들이 원하는 k_A와 k_B의 덧셈 형태를 만들 수 없다. 따라서 사용자 간의 공모로 유효한 가명을 만들어낼 수 없다.
- **다수의 유효한 가명의 생성 방지:** 어떤 사용자는 자신이 얻은 정보를 이용하여 적법하지 않지만 유효한 가명을 만들어 서비스 제공자와 판단기관의 추적 가능성을 배제하려고 할 수 있다. 사용자가 다음과 같이 $(m-j)$개의 예비 가명을 조합하는 경우를 살펴보자.

$$C^* = \left((c_1'')^k\times(c_2'')^k\times\cdots\times(c_{m-j}'')^k\right)^r = g^{brk(m-j)}g^{(a_1''+a_2''+\cdots+a_{m-j}'')xrk} \quad (10)$$

$$E^* = \left((e_1'')^k\times(e_2'')^k\times\cdots\times(e_{m-j}'')^k\right)^r = g^{(a_1''+a_2''+\cdots+a_{m-j}'')rk} \quad (11)$$

1) 동일한 c_i와 e_i를 여러 번 사용할 수 있도록 허용하였을 때, 판단기관이 사용자에게 요구하는 s_i의 수가 줄어들게 되므로, 사용자는 서비스 제공자로부터 판단기관이 지정한 $(L_c'',L_e'')\Leftarrow L_v''$이외에 더 많은 c_i와 e_i를 가져올 수 있으며 이를 지정된 가명 E이외에 다른 가명을 만드는데 사용할 수 있다. 이 과정에서 1~5단계를 무사히 통과할 수 있다. 그러나 6단계에서 사용자는 $H(E)$를 전송해야하므로 이렇게 생성한 가명을 사용할 수는 없다. 따라서 복원 추출 방식으로 가명의 수가 결정된다. 이는 비복원 추출 방식으로 결정되는 ${}_nC_m$보다 크다.

[표 2] 단계별 주요 연산량 비교, ()안은 실시간 연산에서 제외할 수 있는 연산

		판단기관	서비스 제공자	사용자
가명의 획득 단계	1단계	서명 1번		
	2단계		서명 확인 1번 제곱 n번 재암호화 1번 OT_n^m 1번	OT_n^m 1번 곱셈 $m-1$번
	3단계			
	4단계	(OT_n^m 1번)	(OT_n^m 1번)	
	5단계	서명 1번 (곱셈 m번) (해시 1번)	서명 확인 1번 제곱 n번	곱셈 $m-1$번
가명의 사용 단계	6단계		제곱 5번 역 1번 곱셈 3번 해시 1번	(제곱 2번) (재암호화 2번)
익명의 취소 단계	7단계	로그 분석 1번		

위의 가명 조합에서 얻을 수 있는 $W^*=g^{brk(m-j)}$와 같다. 식 (6)을 만족해야 하므로, $g^{bmw}=g^{brk(m-j)}$이다. 따라서 $w=rk(m-j)/m$이어야 하는데, $(\alpha_0',\beta_0')=(r((m-j)/m)\alpha_0\alpha_1^r,\beta_0\beta_1^r)$로 메시지를 다시 만들면 식 (6)이 만족하게 된다. 또는 $w=rk$로 만들고 C^*, E^*의 제곱을 $mr/(m-j)$로 하는 것도 가능하다. 그러나 그 후에 생성되는 가명 $P^*=(E^*)^{1/w}$는 E와 다르다.

$$P^*=(E^*)^{1/w}=\left(g^{(a_1''+\cdots+a_{m-j}'')rk}\right)^{\frac{m}{(m-j)rk}} =g^{(a_1''+\cdots+a_{m-j}'')\frac{m}{m-j}} \quad (12)$$

$H(P^*)\neq H(E)$이므로 SP의 인증을 통과하기 위해서는 사용자가 $H(P^*)$를 계산하여 $H(E)$ 대신 (α_2,β_2)를 만들어 보내야 하는데, 사용자가 알고 있는 가명에 대한 정보 $\{c_i^k\}$와 $\{e_i^k\}$는 모두 사용자가 알지 못하는 k에 의해서 제곱 되어 있어 $H(P^*)$를 계산하기 어렵다. 따라서 사용자는 자신이 알고 있는 예비 가명에 대한 정보를 이용하여 여러 개의 가명을 생성할 수 없다.

4.4 전체 연산량과 실시간 연산을 위한 최적화

이 프로토콜을 수행하기 위하여 각 참여자가 수행해야 하는 주요 연산의 양은 단계별로 [표 2]과 같다. [표 2]에서 알 수 있다시피, 가장 큰 연산은 OT와 깊이 관련 되어 있다. 연산량을 줄이고자 한다면 보안 파라미터 m과 n의 크기를 줄이는 것이 좋다. 그러나 이 보안 파라미터의 크기를 줄일 경우 사용자의 익명성이 낮아지는 결과를 가져오게 된다. 따라서 성능과 보안 측면을 고려하여, 각 응용 환경에 맞는 보안 파라미터를 정하는 것이 중요하다.

실시간으로 일어나는 연산을 줄이기 위해서, 판단기관과 서비스 제공자는 프로토콜에 필요한 연산을 미리 수행해놓을 수 있다. 판단기관은 사용자들을 위하여 미리 L_v'와 L_v''를 선택해놓고 $H(E)$를 계산해놓을 수 있다. 따라서 4단계에서 판단기관과 서비스 제공자 사이에서 발생하는 OT의 연산을 사용자와 관련 없이 미리 수행해놓을 수 있다는 것과 같다. 사용자는 자신이 사용할 가명에 대해서 미리 연산을 수행해 놓을 수 있다.

기존 OT 기반 가명 획득 프로토콜[14]과 비교하면, 제안하는 프로토콜은 실시간 연산에 대해서 판단기관이 실시간으로 수행해야했던 $m-1$번의 복호화 연산이 사라졌고, 유휴시간에 수행할 수 있는 OT 연산이 추가되었다. 서비스 제공자의 경우 기존에는 없었던 OT 연산과 가명 확인을 위한 연산이 추가되었으며, 예비 가명에 대한 변형 연산에 의해서 $2n$번의 제곱 연산이 늘어났다. 사용자의 입장에서 m번의 암호화 연산이 사라지고 대신 $2(m-1)$번의 곱셈을 수행해야 한다. 전체적으로 보았을 때, 제안하는 프로토콜은 기존 OT 기반 가명 획득 프로토콜에 비하여 사용자와 판단기관의 실시간 연산의 부담이 줄어들었으며, 서비스 제공자의 연산량이 다소 늘어났음을 알 수 있다.

4.5 익명성의 제한

사용자의 익명성이 제한되는 경우는 사용자의 익명성보다 높은 가치의 무엇이 발생하였을 경우이다. 이 프로토콜에서 사용자의 익명성은 두 참여자, 즉 판단기관과 서비스 제공자가 모두 동의하는 '사용자의 익명성보다 높은 가치'가 발생하였을 때이기 때문에 서비스 제공자가 독단적으로 판단하는 것보다 사용자의 익명성이

잘 보장되는 특징이 있다.

서비스 제공자는 판단기관의 협력을 통해 어떤 익명 사용자의 실명을 알아낼 수 있다(조건부 추적). 서비스 제공자와 판단기관의 협력은 통신을 통해 완전히 자동으로 일어날 수도 있지만, 각각 기관의 관리자들이나 담당자들 사이에서 전화나 서면을 통해 일어날 수도 있다. 서비스 제공자는 이후의 해당 사용자로부터 요청되는 가명 획득 절차 자체를 거부하거나 사용자에 대한 유예기한을 둘 수 있을 것이다.

서비스 제공자와 판단기관은 다른 형태의 협력을 통하여 완전한 실명 노출을 통하지 않고도 특정 사용자를 시스템에서 제한할 수도 있다. 서비스 제공자는 단독으로 특정 가명의 접근을 제한할 수 있다. 그러나 서비스 제공자는 사용자의 실명을 모르기 때문에 해당 사용자는 새로운 가명을 획득하여 서비스 제공자에게 계속 서비스를 요청할 수도 있다. 이러한 상황을 막기 위하여 서비스 제공자는 특정 가명의 접근을 제한하는 동시에 판단기관에 특정 가명을 전송하여 해당 가명 사용자에 대한 새로운 가명 획득 절차를 (일정 기간 또는 영구히) 시작하지 못하도록 요청해야 한다.

Ⅴ. 결 론

이 논문에서는 기존의 OT를 이용한 가명 획득 프로토콜의 여러 장점을 유지한 채, 예비 가명들을 조합하여 가명을 생성하고 이를 사용자가 서비스 제공자에게 사용하는 방법을 제안하였다. 이렇게 함으로써, 서비스 제공자는 가명의 고갈에서 크게 자유로울 수 있었다. 서비스 제공자는 사용자의 유효한 가명이 요청되었을 때에 가명에 대한 레코드를 준비하면 되므로, 가명 풀과 익명 사용자의 레코드를 분리하여 관리할 수 있게 되었다. 예비 가명을 조합해서 가명으로 사용하도록 하였을 때 발생할 수 있는 사용자간에 예비 가명을 이용한 공모나 복수의 예비 가명을 만들 수 있는 문제를 재암호화와 일방향 함수의 이용을 통해 해결할 수 있었다.

제안하는 프로토콜은 기존의 OT 기반 가명 획득 프로토콜과 마찬가지로 익명성을 필요로 하지만 가명의 연속성이 필요로 하는 분야에 사용될 수 있다. 익명의 토론 게시판이나 익명의 상담 등에 사용될 수 있다. 물론, 외부에 노출되는 닉네임과 같은 가명은 사용자에 의해서 바뀔 수 있지만, 서비스 제공자는 여전히 해당 인물이 동일한 가명을 사용하고 있음을 알 수 있다. 사용자의 잘못된 행동 등이 벌어졌을 때, 서비스 제공자는 해당 가명을 차단하거나 판단기관과 협력하여 해당 사용자의 실명을 추적할 수 있는 기능도 가지고 있다.

참고문헌

[1] D. Chaum, “Security without identification transaction systems to make Big Brother obsolete,” Communications of the ACM, vol. 28, no. 10, pp. 1030-1044, Oct. 1985.

[2] D. Chaum and J.H. Evertse, “A secure and privacy protecting protocol for transmitting personal information between organizations,” CRYPTO’86, LNCS 263, pp. 118-167, 1986.

[3] I.B. Damgard, “Payment systems and credential mechanisms with provable security against abuse by individuals,” CRYPTO’88, LNCS 403, pp. 328-335, 1988.

[4] L. Chen, “Access with pseudonyms,” Cryptography: Policy and Algorithms, LNCS 1029, pp. 232-243, 1995.

[5] A. Lysyanskaya, R.L. Rivest, A. Sahai, and S. Wolf, “Pseudonym systems,” Workshop on Selected Areas in Cryptography 1999, LNCS 1758, pp. 184-199, 1999.

[6] J. Camenisch and A. Lysyanskaya, “An efficient system for non-transferable anonymous credentials with optional anonymity revocation,” EUROCRYPT’01, LNCS 2045, pp. 93-118, 2001.

[7] M. Layouni and H. Vangheluwe, “Anonymous k-show credentials,” EuroPKI’07, LNCS 4582, pp. 181–192, 2007.

[8] P.P. Tsang, M.H. Au, A. Kapadia, and S.W. Smith, “Blacklistable anonymous credentials: blocking misbehaving users without TTP,” ACM Conference on Computer and Communications Security 2007, pp. 72-81, Oct. 2007.

[9] V. Benjumea, J. Lopez, J.A. Montenegro, and J.M. Troya, “A First Approach to Provide Anonymity in Attribute Certificates,” PKC 2004,

LNCS 2947, pp. 402-415, 2004.
[10] 권태경, 박해룡, 이철수, “공개키 기반 구조에 기반한 익명게시판 기술 현황,” 정보보호학회지, 14(6), pp. 1-13, 2004년 12월.
[11] T. Kwon, J.H. Cheon, Y. Kim, and J. Lee, “Privacy Protection in PKIs: A Separation-of-Authority Approach,” International Workshop on Information Security Applications, LNCS 4298, pp. 297-311, 2007.
[12] G. Brassard, C. Crépeau, and J.M. Robert, “All-or-nothing disclosure of secrets,” CRYPTO’86, LNCS 263, pp. 234 – 238, 1986.
[13] Y. Mu, J. Zhang, and V. Varadharajan, “m out of n oblivious transfer,” Australasian Conference on Information Security and Privacy, LNCS 2384, pp. 395-405, 2002.
[14] 양대헌, 이경희, “추적 가능한 가명 은밀 획득 프로토콜,” 정보보호학회논문지, 16(5), pp. 113-118, 2006년 10월.
[15] M. Jakobsson, A. Juels, and P. Syverson, “Universal re-encryption for mixnets,” RSA Conference 2004, Cryptographer’s track, LNCS 2964, pp. 163-178, 2004.

<著者紹介>

강 전 일 (Jeonil Kang) 정회원
2003년 2월: 인하대학교 컴퓨터 공학과 졸업
2006년 2월: 인하대학교 정보통신대학원 석사
2006년 3월~현재: 인하대학교 정보공학과 박사 과정
<관심분야> RFID 보안, 생체 인식 보안, 무선 센서 네트워크 , 무선 인터넷 보안, 웹 인증 보안

양 대 헌 (DaeHun Nyang) 종신회원
1994년 2월: 한국과학기술원 과학기술 대학 전기 및 전자 공학과 졸업
1996년 2월: 연세대학교 컴퓨터 과학과 석사
2000년 8월: 연세대학교 컴퓨터 과학과 박사
2000년 9월~2003년 2월: 한국전자통신연구원 정보보호연구본부 선임연구원
2003년 2월~현재: 인하대학교 정보통신대학원 조교수
<관심분야> 암호이론, 암호프로토콜, 인증프로토콜, 무선 인터넷 보안

이 경 희 (KyungHee Lee) 정회원
1989년: 서울대학교 식품영양학과 학사
1993년: 연세대학교 전산과학과 학사
1998년: 연세대학교 컴퓨터과학과 석사
2004년: 연세대학교 컴퓨터과학과 박사
1993년 1월~1996년 5월: LG소프트(주) 연구원
2000년 12월~2005년 2월: 한국전자통신연구원 선임연구원
2005년 3월~현재: 수원대학교 전임강사
<관심분야> 영상처리, 컴퓨터비젼, 인공지능, 패턴인식, 생체인식, 얼굴인식, 다중생체인식