情報保護學會論文誌
第 21 卷 第 5 號, 2011. 10

웹 환경에서 익명성을 제공하는 자격증명 방법

이 윤 경,† 황 정 연, 정 병 호,‡ 김 정 녀
한국전자통신연구원 지식정보보안연구부

Anonymous Qualification Verifying Method on Web Environment

Yun-kyung Lee,† Jung Yeon Hwang, Byung Ho Chung,‡ Jeong Nyeo Kim
Electronics and Telecommunications Research Institute

요 약

인터넷 이용자의 개인정보 유출 및 인터넷에서의 행적이 연결되는 등의 프라이버시 침해에 관한 논란이 끊이지 않고 있으며, 이를 해결하기 위한 다양한 방법들이 제시되고 있지만, 가장 효율적인 개인정보보호 방법으로 익명성을 제공하는 인증, 접근제어, 지불 메커니즘을 들 수 있다. 따라서, 본 논문에서는 사용자의 익명성을 보장하면서 인증을 하고 이와 함께 접근제어를 할 수 있는 효과적인 방법을 제안한다. 제안 기법은 핵심 프리미티브로 그룹서명 기법을 이용하며, 접근제어를 위한 익명자격증명서를 도입하고, 관련된 익명 인증 및 자격정보를 상호 결합한다. 정당한 사용자는 키 발급 절차를 통해 합법적으로 그룹 서명키를 얻은 후 자신의 자격에 대한 익명자격증명서를 발급받고, 이를 이용함으로써 익명 인증 및 접근제어를 제공하는 웹서비스를 안전하고 편리하게 사용할 수 있다. 익명자격증명서는 응용환경에 따라 유연하게 확장하여 사용할 수 있으며 익명자격증명서를 지불토큰으로 이용하여 익명 지불 기법에 활용할 수도 있다.

ABSTRACT

There's a controversy about an invasion of privacy which includes a leakage of private information and linking of user's behavior on internet. Although many solutions for this problem are proposed, we think anonymous authentication, authorization, and payment mechanism is the best solution for this problem. In this paper, we propose an effective anonymity-based method that achieves not only authentication but also authorization. Our proposed method uses anonymous qualification certificate and group signature method as an underlying primitive, and combines anonymous authentication and qualification information. An eligible user is legitimately issued a group member key pair through key issuing process and issued some qualification certificates anonymously, and then, he can take the safe and convenience web service which supplies anonymous authentication and authorization. The qualification certificate can be expanded according to application environment and it can be used as payment token.

Keywords: anonymous access control, anonymous authentication, privacy protection

I. 서 론

인터넷이 활성화됨에 따라 인터넷은 우리 생활에 없어서는 안 될 중요한 요소로 자리 잡고 있지만, 이와 함께 실제 (물리적) 생활에서는 중요하게 고려되지 않았던 사용자 프라이버시 침해 문제가 크게 부각되고 있다. 즉, 인터넷 이용자의 개인정보 유출 및 인터넷에서의 사용자 행적 연결로 인한 프라이버시 침해는 심각한 문제가 아닐 수 없다. 최근 잇따라 발생한 대형 포탈 및 금융권의 개인정보 유출사고 뿐만 아니라,

접수일(2011년 7월 29일), 게재확정일(2011년 10월 8일)
† 주저자. neohappy@etri.re.kr
‡ 교신저자. cbh@etri.re.kr

크고 작은 사회적 이슈가 되는 사건에 대해서 사건 당사자의 신상 털기는 인터넷의 새로운 문제로 대두되고 있는 실정이다. 또한 인터넷의 특성상 한 번 노출된 정보는 삽시간에 수천, 수만의 사람들에게 퍼져나갈 수 있다는 사실은 더욱 큰 문제가 아닐 수 없다. 그러나 대부분의 인터넷 서비스제공자들은 서비스 이용 시 회원가입을 요구하고 있고, 회원 가입 시 개인정보의 입력을 요구하고 있으며, 이렇게 수집된 개인 정보에 대한 보안은 그다지 철저하지 못하다. 이러한 상황에 대해서 많은 인터넷 이용자들은 개인정보 보호 및 프라이버시 보호에 대한 요구를 높이고 있으며 인터넷으로 하는 모든 행동들이 수집되고 기록될 수 있다는 문제에 대해 특별한 관심을 보이고 있고, 익명성의 필요성을 공감하고 있다[1].

익명성을 이용한 다양한 인증 방법이 제시되어 있고, 이들 기술을 실제 인터넷 서비스에 적용하여 사용자 프라이버시를 보호할 수 있을 것이다. 하지만, 실용적인 관점에서는 보다 폭넓은 기능을 제공하는 프라이버시 보호 기반의 응용 방법이 요구된다. 예를 들어, 익명으로 인증하고 서비스를 제공할 경우, 성인만 접근할 수 있는 서비스라든지 특정 자격을 가진 사람만이 접근할 수 있는 서비스에 대한 사용자 접근제어가 힘들어진다. 따라서 서비스제공자는 익명으로 사용자에게 서비스를 제공하면서 접근제어도 함께 할 수 있는 방안이 마련되지 않는다면 익명 인증기술을 실제 서비스에 적용하기를 꺼려할 것이다. 사용자가 자신의 자격(qualification)을 증명하여 그 자격에 따라 접근제어를 하는 방법들은 이전부터 논의되어 왔지만 익명성의 관점에서 많은 단점들을 가지고 있다.

1.1. 논문의 결과(Contributions)

본 논문에서는 사용자의 프라이버시 보호를 위해, 사용자가 노출하고자 하는 정보만을 서비스제공자에게 제출하여 서비스에 대한 접근제어(access control)를 할 수 있도록 하는 인가방법을 제안한다.

제안 기법은 그룹서명[7]을 이용하는 익명 인증시스템에 기반한다. 익명 인증 및 인가 서비스를 위해 사용자는 유효한 그룹 멤버키를 발급 받은 후 인증 및 인가를 위한 그룹 서명을 생성하는데 이를 이용한다. 익명성을 제공하는 속성 기반의 접근제어 서비스를 위해, 사용자는 그룹 서명과 결합된 자신의 속성을 증명함으로써 특정 서비스에 대한 접근권한을 얻어낸다. 효과적인 속성의 표현을 위해, 알려진 PMI(Privilege Management Infrastructure Certificate) 인증서의 구조를 따르되, 기존의 PMI 인증서에서 발생하는 사용자의 실명정보 노출 문제를 해결하기 위해, 본 논문에서는 PMI 인증서의 소유자 필드 값으로 사용자의 완전한 익명정보를 이용한다. 제안 방법은 하나의 웹서비스 안에서는 물론이고, 웹서비스간 에서도 사용자의 행적을 연결할 수 없으므로 사용자의 프라이버시가 보호된다는 장점이 있다.

자연스러운 확장을 통해 본 논문에서 제안한 익명 자격증명서를 지불토큰으로 활용할 수 있다. 따라서 본 제안 기법을 활용하여 익명성 기반의 인증, 인가 및 지불이 가능한 시스템의 구현이 가능하다. 제안 기법은 하나의 그룹 멤버키로 서비스를 이용할 수 있으므로 서비스 이용이 간편하고, 사용자의 선택에 따라서 인증만 하거나 인증과 인가를 동시에 할 수 있으므로 서비스 이용의 유연성을 제공할 수 있다.

1.2. 관련 연구(Related Works)

1.2.1. 익명인증

암호학적 관점에서, 익명성을 제공하는 프리미티브의 연구는 다양하게 진행되고 있다. 익명성을 지원하는 대표적인 서명 방법으로 그룹 서명(group signature)[3], 링 서명(ring signature)[21] 및 추적 가능한 서명(traceable signature)[2]을 꼽을 수 있다. 특히 1991년 D.Chaum과 V.Heyst가 소개한 그룹 서명[3]은 사용자(signer)가 자신의 신원을 노출하지 않으면서 그룹에 속해 있음을 증명할 수 있는 기술로써, 서명을 검증하는 쪽(verifier)에서는 특정 그룹의 멤버 중 한 사람이 서명을 했고, 그 서명이 유효함을 알 수 있지만, 그룹 멤버 중 누가 그 서명을 했는지에 대해서는 알 수 없기 때문에 익명성이 보장된다고 할 수 있다. 다만, 그룹 마스터키를 소유한 그룹 매니저(group manager)만이 서명 값으로부터 서명을 생성한 그룹 멤버를 알아 낼 수 있다. 링 서명은 키를 가진 멤버들은 모두 익명으로 서명이 가능하지만, 그룹 매니저의 개념이 없기 때문에 서명 생성자를 알 수 없으며, 멤버 탈퇴(revocation) 기능을 지원하지 않는다는 점에서 그룹서명과 차이가 있다. 또한 추적 가능한 서명은 그룹서명처럼 그룹 매니저가 서명 값으로부터 서명한 사람을 알 수 있지만, 서명자 본인 또한 원하는 경우 특정 서명에 대해서 자신이 서명한 것임을 밝힐 수 있다는 점에서 그룹서명과 차이

가 있다.

1991년 D. Chaum과 V. Heyst가 그룹 서명[3]을 소개한 이후, 그룹 서명을 이용한 익명인증 기법에 대한 연구가 최근까지 활발히 진행되고 있다. 초기의 그룹 서명은 그룹 멤버의 수에 비례하여 서명의 길이 또한 증가하는 단점이 있어서 서비스에 적용하기에는 한계가 있었다. 이러한 문제를 해결하기 위해 J. Camenisch와 M.Stadler[4], G. Ateniese et al.[5], L. Nguyen과 R.Safavi-Naini[6]는 그룹 크기와 상관없이 고정된 길이의 그룹 공개키와 서명을 가지는 그룹 서명 기법을 제안하였다. 2004년 D. Boneh et al.은 1024비트 RSA 서명과 동일한 안전성을 가지고, 대략 비슷한 길이의 서명을 생성하는 (171 비트 사이즈의 엘리먼트를 갖는 G_1 그룹(170 bits prime order)을 이용할 경우 1533비트) 짧은 그룹 서명 기법을 제안[7] 하였는데, 기존의 대부분의 그룹 서명이 Strong RSA 가정에 기반한 것과는 달리 D. Boneh et al.은 이선형함수(bilinear map)를 가진 그룹에서의 Strong Diffie-Hellman 가정[8]을 이용함으로써 비교적 짧은 길이의 그룹서명을 얻을 수 있었다.

1.2.2. 익명인가

Kiyomoto et al.은 속성 인증서(attribute certificate)를 발급받아서 숨김 서명(blind signature)을 한 후 서비스제공자에게 제출함으로써 추적 불가능한 익명 인증 및 인가가 가능함을 보여주었고[9], Benjumea et al.은 가명(pseudonym)을 생성하여 속성인증서(PMI 인증서)의 소유자(holder) 필드에 넣고, 이를 이용하여 익명 인증 및 인가를 받는 구조를 제시하였다[10]. 그러나 사용자가 이 익명 PMI인증서를 이용하여 서비스를 받게 되면, 사용자의 웹상에서의 행적이 모두 연결(link)되어 익명성을 해치게 될 수 있다는 단점이 있다. 또한 신수연 등은 D. Boneh et al.의 그룹서명[7]을 변형하여 익명으로 인가가 가능한 그룹서명 기법을 제시[11, 12] 하였는데, 그룹 멤버키 발급 시 사용자의 권한정보를 그룹 매니저에 제출하여 그룹 멤버키에 사용자의 권한정보를 삽입하는 방법을 이용하고 있다. 이 경우, 사용자는 자신의 권한정보의 개수만큼 그룹 멤버키를 발급받아야 하므로 사용자는 여전히 여러 개의 그룹 멤버키를 관리하여야 하는 문제를 안고 있고, 익명 인증만으로 이용 가능한 서비스를 이용하고자 할 때, 서비스제공자에게 사용자의 권한을 불필요하게 알려주게 된다는 단점이 있다. M. Backes et al.은 자신의 인증서(Certificates)와 속성(attribute)을 암호화하여 서비스제공자에게 제시하고, 서비스제공자는 사용자의 인증서와 속성을 복호화 하여 해당 서비스를 이용할 수 있는 권한이 있는 사용자인지를 체크한 후 사용자에게 서비스 제공여부를 결정하는 익명 인가 기법을 제시하였다[13]. Ren et al.은 모바일 환경에서 적용 가능한 익명성 기반의 인증 및 인가구조를 제시하였는데, 사용자가 신용장(credential)을 생성하여 서비스제공자에게 제시하고, 이를 검증함으로써 서비스제공자는 사용자에 대한 인증 및 인가가 가능한 구조[14]이지만, 최근 C. Li et al.은 Ren et al.이 제안한 프로토콜이 사용자가 자신의 권한을 남용하는 것이 가능하다는 의견을 제시하였다[15]. K.Shin et al.은 서비스제공자가 자신의 서비스에 대한 접근권한을 나타내는 티켓을 사용자에게 발급하고, 사용자는 이 티켓으로 서비스제공자의 서비스를 이용하기 때문에 서비스제공자가 사용자에 대해서 알고 있는 정보는 제한되고, 사용자가 새로운 티켓을 발급받아서 서비스를 이용하게 되면 서비스제공자는 사용자의 행적을 링크할 수 없다는 점에서 익명성을 제공할 수 있는 접근제어 기법을 제안하였다[16, 17]. 사용자가 자신의 자격(qualification)을 증명하여 그 자격에 따라 접근제어를 하는 방법은 이전부터 논의되어 왔으며 대표적인 예로 2001년 K. Shin이 제안한 방법[18]과 2003년 N. Li et al.이 제안한 방법[19]을 들 수 있다. 또한 이러한 ABAC(Attribute Based Access Control) 방법은 접근제어를 위한 새로운 접근 방법으로 각광받고 있다.

1.2.3. 지역 연결성을 제공하는 그룹 서명 기법[20]

최근 Boneh, Boyen, Shacham의 짧은 그룹 서명(Short Group Signature) 기법[7]을 확장하여 지역연결성(Local Linkability, LL)을 제공하는 그룹 서명 기법이 [20]에서 제안되었다. 본 장에서는 이에 대해 간단히 살펴본다. 편의상 [20]의 기법을 SGS-LL로 부르기로 한다. SGS-LL 기법은 본 논문의 제안 기법을 위한 주요 핵심 프리미티브 중 하나로 이용될 예정이다.

SGS-LL 기법은 그룹 멤버 키 관리 기능을 수행하는 멤버키 발급자(Issuer)와 서명으로부터 서명자를 알아낼 수 있는 멤버 확인자(Opener), 서명자,

그리고 서명 검증자의 네 요소로 구성된다. 멤버키 발급자(Issuer)는 사용자의 그룹 멤버키를 생성 및 발급하고, 또한 폐기를 담당한다. 멤버 확인자(Opener)는 서비스제공자가 서명을 생성한 사람에 대한 공개를 요청할 때 서명자 확인키 (opening key)를 이용하여 서명값으로부터 서명자의 신원을 알아낸다. SGS-LL은 [7]에 Link 알고리즘을 추가하여 확장한 형태로써, SGS-LL의 안전성은 [7]과 동일한 안전성을 갖는다. 보다 자세한 내용은 [20]을 참조한다.

SGS-LL은 이선형 군 쌍(G_1, G_2, G_T)과 이와 결합된 이선형 함수(bilinear map) $e: G_1 \times G_2 \rightarrow G_T$를 이용하며 다음과 같이 KeyGen, Sign, Verify, Open, Revoke, Link의 여섯 알고리즘들로 구성된다. 다음에서 그룹전체 맴버 수는 n 이라고 가정하자.

- 키 생성 알고리즘 KeyGen은 그룹 공개키, 그룹 멤버들의 비밀키, 서명자 확인키 (opening key)를 생성하는 과정이다. 먼저 G_1에서 h를 랜덤으로 선택하고, Z_p^*에서 ξ_1, ξ_2를 랜덤으로 선택한다. 이때 (ξ_1, ξ_2)는 서명자 확인키 (opening key)로 멤버 확인자가 관리하는 비밀키이다. 멤버키 발급자(Key Issuer)는 $u^{\xi_1} = v^{\xi_2} = h$를 만족하는 u, v를 G_1에서 선택하고, Z_p^*에서 γ를 임의로 선택하여, $w = g_2^{\gamma}$ 를 계산한다. γ는 맴버키 발급키 (issuing key)로 멤버키 발급자가 관리하는 비밀키이다. γ를 이용하여 $1 \le i \le n$인 각 그룹 멤버 i에 대해서 그룹 멤버의 비밀키 (A_i, x_i)를 생성하고, 그룹 공개키 $gpk = (g_1, g_2, h, u, v, w)$를 공개한다.
- 서명 생성 알고리즘 Sign은 그룹 공개키 gpk와 그룹 멤버의 비밀키 $gsk[i] = (A_i, x_i)$, 서명할 메시지 $M \in \{0,1\}^*$에 대해서 다음과 같이 서명값을 생성한다. 먼저 Z_p에서 α, β를 임의로 선택하고, $T_1 \leftarrow u^{\alpha}$, $T_2 \leftarrow v^{\beta}$, $T_3 \leftarrow A \cdot h^{\alpha+\beta}$가 되는 선형 암호화(linear encryption)를 수행하고, $\delta_1 \leftarrow x\alpha$, $\delta_2 \leftarrow x\beta$를 만족하는 δ_1과 δ_2를 연산한다. 또한 Z_p에서 $r_{\alpha}, r_{\beta}, r_{\gamma}, r_{\delta_1}, r_{\delta_2}$를 임의로, 그리고 고른 분포를 따라 (uniformally at random) 선택한 후, 다음과 같이 R_1, R_2, R_3, R_4, R_5를 연산한다:

$R_1 \leftarrow u^{r_{\alpha}}$, $R_2 \leftarrow v^{r_{\beta}}$,

$R_3 \leftarrow e(T_3, g_2)^{r_x} \cdot e(h, w)^{-r_{\alpha}-r_{\beta}} \cdot e(h, g_2)^{-r_{\delta_1}-r_{\delta_2}}$

$R_4 \leftarrow T_1^{r_x} \cdot u^{-r_{\delta_1}}$, $R_5 \leftarrow T_2^{r_x} \cdot v^{-r_{\delta_2}}$.

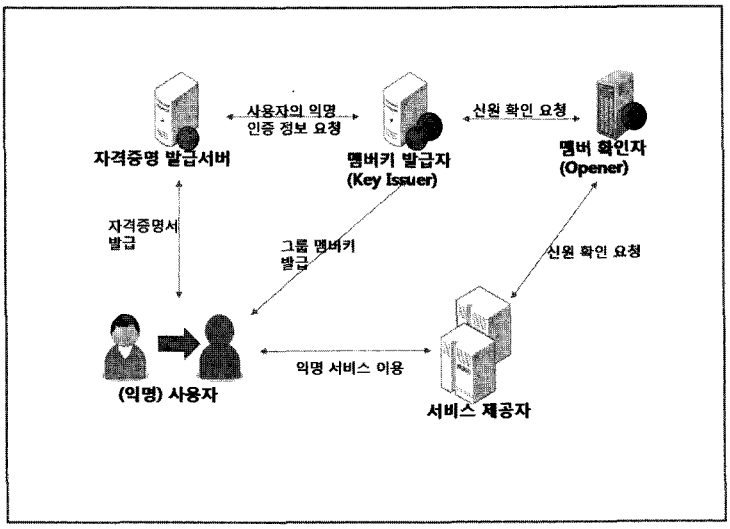
그리고 서명할 메시지 M과 앞서 연산한 T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5를 이용하여 해쉬값 $c \leftarrow H(M, T_1, T_2, T_3, c, s_{\alpha}, s_{\beta}, s_x, s_{\delta_1}, s_{\delta_2})$를 생성하고 다음과 같이 $s_{\alpha}, s_{\beta}, s_x$, $s_{\delta_1}, s_{\delta_2}$값을 연산한다:

$s_{\alpha} \leftarrow r_{\alpha} + c\alpha, s_{\beta} \leftarrow r_{\beta} + c\beta$,

$s_x \leftarrow r_x + cx$, $s_{\delta_1} \leftarrow r_{\delta_1} + c\delta_1$,

$s_{\delta_2} \leftarrow r_{\delta_2} + c\delta_2$.

서명값 $\sigma = (T_1, T_2, T_3, c, s_{\alpha}, s_{\beta}, s_x, s_{\delta_1}, s_{\delta_2})$을 출력한다.



[그림 1] 익명성을 제공하는 속성 기반 접근제어 시스템 프레임워크

- 서명 검증 알고리즘 Verify는 그룹 공개키 gpk와 메시지 M, 이에 대응하는 그룹 서명 값 $\sigma=(T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$를 입력으로 받아서 서명 값이 유효한지 검증한다. 먼저 아래 수식과 같이 R_1, R_2, R_3, R_4, R_5를 연산한다:

  $R_1 = u^{s_\alpha} T_1^{-c}$, $R_2 = v^{s_\beta} T_2^{-c}$,

  $R_3 = e(T_3, g_2)^{s_x} \cdot e(h, w)^{-s_\alpha - s_\beta} \cdot$

  $\cdot\ e(h, g_2)^{-s_{\delta_1} - s_{\delta_2}} \cdot \left(\frac{e(g_1, g_2)}{e(T_3, w)}\right)^{-c}$,

  $R_4 = T_1^{s_x} u^{-s_{\delta_1}}$, $R_5 = T_2^{s_x} v^{-s_{\delta_2}}$.

  그리고 $c' \leftarrow H(M, T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$을 계산한다.

  만일 등식 $c'=c$이 성립하면 "*유효함*"을 출력하고, 그렇지 않으면 "*유효하지 않음*"을 출력한다.
- 서명자 확인 알고리즘 Open은 그룹 공개키 gpk와 서명자 확인키(opening key) $mok=(\xi_1, \xi_2)$, 메시지 M, 서명 값 $\sigma \leftarrow (T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$를 이용하여 멤버 확인자(Opener)가 서명 값 σ를 생성한 그룹 멤버를 찾는 과정이다. 보다 구체적으로, 주어진 유효한 서명에 대해 $A_i \leftarrow T_3 \cdot (T_1^{\xi_1} \cdot T_2^{\xi_2})^{-1}$ 을 계산한 후 등록된 그룹 멤버 비밀키 목록의 항목들과 비교하여 일치하는 A_i값을 찾아냄으로써 해당 서명자를 찾을 수 있다.
- 멤버 철폐 알고리즘 Revoke는 멤버키 발급자(Key Issuer)가 멤버 자격을 박탈(revoke)할 그룹 멤버들의 비밀키들로 구성된 멤버 철폐 리스트(Revocation List:RL)를 생성해서 공개하고, 모든 서명자와 서명 확인자가 RL을 통해서 자신의 그룹멤버키를 업데이트하는 알고리즘이다. 자세한 내용은 [7], [20]을 참조한다.
- 연결 알고리즘 Link는 서명 연결키 (linking key) $d=\xi_1/\xi_2$을 이용하여 주어진 각 유효한 그룹서명 $\sigma=(T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$로부터 $e(A, v)=e(T_3, v) \cdot e(T_1^d \cdot T_2, h)^{-1}$를 계산한 후 비교하여 연결 여부를 확인한다. $e(A, v)$ 값은 그룹 멤버들의 멤버 키 폐기(revoke)가 일어나지 않는 동안은(그룹 멤버들 중 한 명이라도 그룹 멤버키의 폐기가 일어나면 모든 멤버의 그룹 멤버키 A가 갱신된다) 동일한 값으로 유지되는 특성에 의해서 서명 값들에 대한 연결이 가능하다.

SGS-LL에서는 실제 웹서비스에 적용할 때, 멤버 확인자(Opener)가 서비스제공자에게 고유한 트랩도어(trapdoor)를 제공하고, 서비스제공자는 이를 이용함으로써, 여전히 사용자의 신원을 알 수 없지만 사용자의 인증 행위를 연결(linkability) 할 수 있는 방안을 제시하였다. 즉, 멤버 확인자(Opener)가 서비스제공자마다 다른 서명자 확인키 (opening key) ξ_1, ξ_2를 생성하고 서명 연결키 (linking key) $d=\xi_1/\xi_2$를 계산한다. 그리고 서비스제공자마다 다른 그룹 공개키 u, v, h(단, $u^{\xi_1}=v^{\xi_2}=h$)를 생성한다. 이러한 구조를 통하여, SGS-LL에서는 다른 서비스 제공자들이 서로 공모를 통하여 서명 값들의 연결 여부를 확인할 수 없음을 보였다.

잘 알려진 바와 같이, SGS-LL과 같은 그룹 서명 기법을 시도 응답식 프로토콜과 결합하여 자연스럽게 익명성을 제공하는 인증 시스템을 구축할 수 있다. 즉, 그룹의 멤버가 서명을 했다는 사실만을 확인 할 수 있고, 그룹 멤버 중 누가 한 서명인지를 알 수 없다. 다만, 만일 정당한 사용자가 그룹 내에서 허용되지 않는 행동을 했을 경우, 멤버 확인자(Opener)가 서명한 사람의 신분을 밝힐 수 있도록 함으로써 불법적인 행동을 한 사람을 추적할 수 있다. 다음 장에서는 이러한 제어 가능한 익명 인증 특성을 확장하여 익명성을 제공하는 속성 기반 접근제어 기법을 구성하는 방법을 제시한다.

II. 익명성을 제공하는 자격 증명 기법

웹서비스를 이용하는데 있어서, 익명 인증만으로는 익명성 기반의 실용적인 서비스를 제공하는데 한계가 있다. 다양한 서비스들은 폭넓은 분야에서 다양한 개체 특성 또는 자격 정보에 기반하여 서비스 되고 있기 때문이다. 예를 들어, 특정회사 직원들만을 위한 서비스, 특정 동호회 회원들만을 위한 서비스, 성인에게만 제공 가능한 서비스 등 원활한 서비스의 제공을 위해서는 인가기능이 필요하다. 이러한 상황을 다루기 위해서, 앞서 기술한 익명 인증 방법을 단순하게 확장하여 이용할 수 있다. 즉, 각 속성에 대응하는 각 그룹을 만들고, 사용자는 자신과 관련된, 자신이 서비스를 받고자 하는 종류의 그룹에서 멤버키를 발급받아 서비스를 이용하는 구조이다. 이 경우, 원하는 기능을 구현할 수 있지만 여러 가지 면에서 많은 단점을 가지고 있다. 먼저 사용자는 자신의 그룹 멤버키를 여러 개 소유하고, 서비스마다 적절한 그룹 멤버키를 선택하여

해당 그룹 멤버키와 관련된 그룹 공개키를 이용하여 서명을 생성하여야 하는 번거로움이 있다. 또한 여러 개의 그룹 멤버키를 이용한다 하더라도 웹 서비스제공자가 사용자에게 제공할 서비스에 대한 세밀한 접근제어를 하기에는 한계가 있다.

위 문제를 해결하기 위한 효과적인 방법으로 익명 자격 증명서를 이용하여 익명성을 제공하는 자격증명 방법을 제안한다. 제안기법을 이용하면 사용자가 익명성을 유지하면서도 특정 자격이 있는 사람만 이용할 수 있는 웹 서비스를 이용하기 위해서 여러 개의 그룹 멤버키를 관리해야 하는 단점을 해결할 수 있고, 사용자가 제시한 익명자격증명서의 자격 내용(속성)에 따라서 서비스에 대한 접근제어를 할 수 있으므로 유연하고 세밀한 접근제어가 가능하다.

2.1. 참가자들(Participants)과 프레임워크 (Framework)

[그림 1]은 익명성 기반의 자격증명 시스템에 대한 전체 프레임워크(framework)를 도식화 한다. 제안기법은 완전한 익명성 제공을 위해서 익명인증 기법의 주요 핵심 프리미티브인 그룹 서명 기법(1.2.3절 참조)을 확장하여 구성된다. 자격 증명을 통해 사용자의 특성에 대한 부가적인 정보 유출이 발생하지 않도록 익명 인증 및 자격 증명을 체계적으로 결합하는 구조를 제공한다. [그림 1]에서 보는 바와 같이, 제안하는 익명성 기반의 자격증명 시스템은 멤버키 발급자(Key Issuer), 멤버 확인자(Opener), 자격증명 발급서버(Qualification Certificate Authority, QCA)의 세 가지 엔터티들과 서명자, 그리고 서명 검증자 (즉, 서비스 제공자)로 구성된다. 상기 시스템에서 멤버키 발급자(Key Issuer), 멤버 확인자(Opener), 자격증명 발급서버(QCA)는 PKI 인증시스템의 CA처럼 신뢰된 엔터티로써 각 엔터티가 소유하고있는 사용자 정보에 대해서 서로간의 공유가 없어야 하고, 또한 사용자 정보에 대한 철저한 보안이 이루어져야 한다. 또한 이들 엔터티들은 각각 독립적인 권한을 가진다.

- **멤버키 발급자(Key Issuer)**: 멤버키를 관리하는 엔터티이다. 최초 사용자 가입 시 사용자의 신원을 확인 한 후 해당 사용자의 멤버키를 발급한다. 멤버키는 SGS-LL 기법을 이용하여 그룹서명을 생성하기 위해 사용되며 익명 자격증명 값과 연계된 유효한 데이터를 생성하기 위해서 필요하다. 멤버키 관리자는 정해진 정책(policy)에 따라 발급한 멤버키를 폐기할 수 있다. 구체적으로 폐기 알고리즘은 [20]의 멤버키 폐기 방법을 따른다. 멤버키 발급자는 사용자와 멤버키를 명시적으로 결합 또는 연계할 수 있지만 생성된 그룹서명으로부터 서명자를 직접 확인할 수는 없다.
- **멤버 확인자(Opener)**: 시스템 초기화 과정 시 주어진 멤버 확인키(opening key)를 이용하여, 주어진 그룹서명 및 이와 연계된 자격 증명 정보로부터 서명자(를 확인해 주거나) 확인에 필요한 정보를 생성하는 엔터티이다. 멤버 확인자는 상기 생성 정보를 이용하여 멤버키 발급자와 협력을 통해 서명자를 확인한다. 사용자의 그룹 멤버키에 대한 멤버 확인자의 접근은 제한된다.
- **자격증명 발급서버 (Qualification Certificate Authority)**: 익명자격증명서를 발급하는 엔터티이다. 익명자격증명서(anonymous qualification certificate) 발급 요청 시, 사용자의 신원을 확인 한 후 자격증명 발급서버는 해당 사용자의 자격을 증명하는 익명자격증명서를 발급한다. 본 제안 모델에서 익명자격증명서 생성은 SGS-LL기법과의 연계를 위해서 자격증명 발급서버와 멤버키 발급자의 협력을 통해 이루어진다. 보다 구체적으로, 익명자격증명서의 소유자 필드(3.4장의 익명자격증명서 구조 참조)에 사용자의 그룹 멤버키와 관련된 정보가 삽입되고, 정보 유출을 방지하기 위해 이 정보는 일회성 난수(random number)로 랜덤화 된다. 자격증명 발급서버는 멤버확인자와 마찬가지로 사용자의 그룹 멤버키 자체를 알지는 못한다.

상기 3개 엔터티들 중 멤버키 발급자(Key Issuer)는 멤버키 발급 시에, 자격증명 발급서버는 익명자격증명서 발급 시에 사용자의 실제 신원 확인 과정이 필요하다. 사용자의 신원 확인 과정은 다양한 방법을 통해 할 수 있다. 예를 들어, 직접 대면을 통해 인증을 할 수도 있고, PKI(Public Key Infrastructure) 인증서를 이용한 실명인증도 가능할 것이다. 본 논문에서는 PKI기반의 X.509 인증서를 이용한 신원 인증을 가정하여 익명자격증명서 발급 프로토콜을 설명한다.

2.2. 보안 요구사항

익명성을 제공하는 자격증명 기법이 기본적으로 가져야 할 보안 특성(security property)으로 사용자간 공모 방지, 서비스제공자간 공모방지, 서비스제공자의 불법행위로 인한 사용자의 익명성 저해 방지를 꼽을 수 있다. 또한 익명자격증명서에 대한 무결성 확보 및 익명자격증명서에 대한 신뢰성(confidentiality) 확보도 필요하다.

- 익명자격증명서 도용방지

익명자격증명서의 도용은 불법으로 취득한 다른 사용자의 익명자격증명서를 자신의 것인 양 사용하는 것을 의미하는 것으로, 사용자가 자신의 비밀키를 다른 사용자에게 알려주어 타인이 대신 그룹서명을 생성할 수 있는 경우는 자신의 권리를 스스로 포기한 것으로 간주하고, 논외로 한다.

- 서비스제공자의 불법행위 방지

익명성을 제공하는 자격증명 기법에서 고려하는 서비스제공자의 불법행위는, 서비스제공자가 사용자의 그룹 멤버키 정보나 사용자들의 행동 패턴을 링크하고자 하는 등 사용자의 익명성을 파괴하는 행동을 하는 것을 의미한다.

- 서비스제공자간 공모 방지

서비스제공자간 공모는 특정 사용자의 행동패턴이나, 우연히 알게 된 사용자의 개인 정보를 서비스 제공 자들 간 서로 공유하는 것을 의미한다. 서비스제공자간 공모가 가능할 경우, 공모에 참여한 서비스제공자들 각각이 알고 있는 사용자의 개인 정보를 조합하여 사용자의 익명성을 저해(파괴)할 수 있기 때문에 서비스제공자간 공모는 불가능하여야 한다.

- 익명자격증명서의 무결성 확보

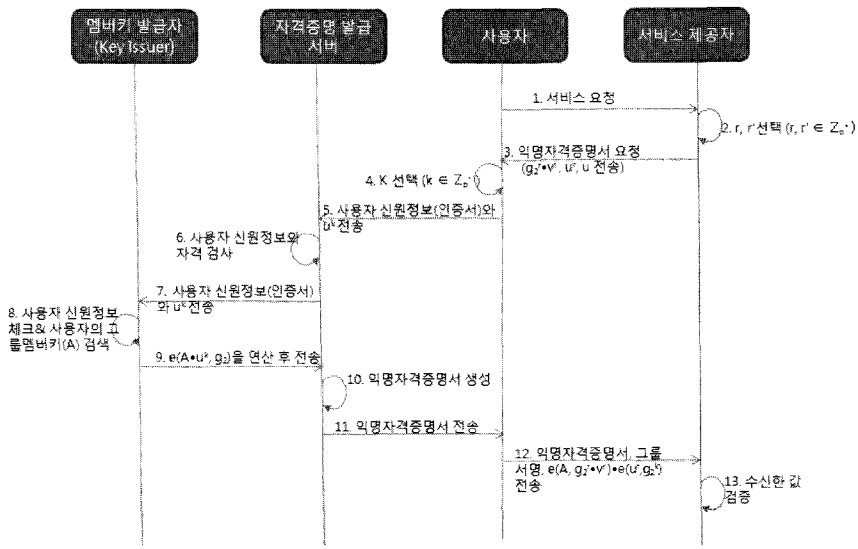
자격증명 발급서버가 한 번 발급한 익명자격증명서는 수정될 수 없어야 함을 의미한다. 익명자격증명서의 발급 이후에 자격증명 발급서버가 아닌 다른 엔터티 혹은 사용자에 의해서 익명자격증명서가 수정되어 사용될 수 있어서는 안 된다.

- 익명자격증명서의 신뢰성 확보

익명자격증명서는 자격증명 발급서버에 의해서 발급된 것임을 확인할 수 있어야 함을 의미한다. 즉, 증명하고자하는 자격에 대해서 자격증명서를 발급할 자격이 있는 기관이 정당하게 발급한 익명자격증명서인지를 확인할 수 있어야 한다.

2.3. 제안 기법

[그림 2]에는 본 논문에서 제안하는 익명성을 제공하는 자격증명 기법을 구체적으로 도시하고 있다. [그림 2]에서 각 엔터티들은 SGS-LL 기법에 따라 익명인증을 이용하기 위해서 필요한 각종 키를 미리 발급받은 상태임을 가정한다. 즉, 사용자는 SGS-LL의 KeyGen 알고리즘을 통해서 자신의 그룹 멤버 비밀키 (A_i, x_i)를 발급 받았고, 서비스제공자는 멤버 확인



[그림 2] 익명성 기반의 인증 및 자격증명 방법

자(Opener)로부터 서명 연결키 (linking key) $d=\xi_1/\xi_2$를 이미 발급 받았다고 가정한다. 또한 키 생성의 하나의 과정인 그룹 공개키 또한 공개되어 있다고 가정한다. 물론 SGS-LL기법에서 제시하였듯이, 그룹 공개키 $gpk=(g_1, g_2, h, u, v, w)$ 중 서비스제공자마다 다른 그룹공개키 u, v, h가 생성되어 공개되어 있다고 가정한다. [그림 2]의 각 과정에서 각 엔터티 간의 통신은 안전한 채널을 통해서 이루어짐을 가정한다. [그림 2]에 제시한 기법을 메시지 흐름에 따라 상세히 설명하면 다음과 같다.

(1) ***사용자 : 서비스 접근 요청***

- 사용자는 서비스제공자에게 익명 인증 및 자격증명과 관련된 서비스 이용(service request)을 요청한다. 본 제안 기법은 익명 인증만을 위한 서비스 제공이 가능하나 여기서는 익명 자격증명이 필요한 서비스(예를들어, 성인만 접근할 수 있는 콘텐츠 접근 서비스)를 가정한다.

(2)~(3) ***서비스제공자 : 사용자에게 익명자격증명서 요청***

사용자가 요청한 서비스에 대해 서비스제공자는 사용자에게 해당 자격을 소유하고 있는 사람인지를 판단하기 위해서 다음과 같이 익명자격증명서를 요구한다.

- $r, r' \in Z_p^*$을 선택하고
- 그룹 공개키 g_2, u, v를 이용하여 $g_2^r \cdot v^{r'}$, u^r을 계산한다.
- [익명자격증명서요청,$(g_2^r \cdot v^{r'}, u^r, u)$]을 사용자에게 전송한다.

(4)~(5) ***사용자 : 자격증명 발급서버에 익명자격증명서 발급 요청***

- 난수 $k \in Z_p$를 생성하고
- 그룹 공개키 u를 이용하여 u^k을 계산한다.
- [익명자격증명서발급요청,자격정보,사용자신원정보(X.509인증서),u^k]을 자격증명 발급서버에 전송한다.

(6)~(7) ***자격증명 발급서버 : 멤버키 발급자(Key Issuer)에게 사용자 익명자격증명 연계 정보 요청***

- 사용자의 신원정보를 체크하고, 사용자의 자격을 검사하여 익명자격증명서를 받을 자격이 있는지를 확인한다.
- 만일 자격이 있는 사람이라면, 자격증명 발급서버는 사용자의 신원정보와 사용자가 전송해 온 u^k을 함께 멤버키 발급자(Key Issuer)에게 전송한다.

(8)~(9) ***멤버키 발급자(Key Issuer) : 자격증명 발급서버에 멤버키와 결합된 익명자격증명 연계 정보 전송***

- 사용자의 신원정보를 확인하고,
- 해당 사용자의 그룹 멤버키 A를 검색하여 사용자가 보내온 u^k과 곱한 후, 이 값과 그룹 공개키 g_2를 페어링 연산한다.
- 페어링 연산 결과값 $e(A \cdot u^k, g_2)$을 자격증명 발급서버에 전송한다.

(10)~(11) ***자격증명 발급서버: 익명자격증명서 생성 및 발급***

- 멤버키 발급자(Key Issuer)가 보낸 $e(A \cdot u^k, g_2)$ 값을 익명자격증명서의 소유자 필드(3.4장 참조)에 넣는다.
- 완전한 익명자격증명서를 생성하여 사용자에게 전송한다.

(12) ***사용자 : 서비스제공자에게 그룹서명과 익명자격증명서 기반의 익명인증 및 자격증명 정보 전송***

- 자신의 그룹 멤버키 A, 서비스제공자가 익명자격증명서 요청시 함께 전송했던 값 $g_2^r \cdot v^{r'}, u^r$ 및 자격증명 발급서버에 익명자격증명서 발급 요청시 사용자가 생성했던 난수 k, 그리고 그룹 공개키 g_2 를 이용하여 $P=e(A, g_2^r \cdot v^{r'}) \cdot e(u^r, g_2^k)$을 연산한다.
- 익명자격증명서에 대한 해쉬값인 H(익명자격증명서)와 상기 계산된 P에 대해서 그룹서명 $\sigma=(T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$을 생성한다.
- 생성한 그룹 서명값 σ와 익명자격증명서, 그리고 P를 서비스제공자에게 전송한다.

(13) ***서비스제공자: 익명인증 및 자격증명 검증***

- 사용자가 전송한 그룹서명을 검증한다.
- 유효한 그룹 서명일 경우, SGS-LL의 연결 알고 리즘 Link를 이용하여 주어진 그룹 서명으로부터 다음과 같이 $O=e(A, v)$를 계산한다.

$$O=e(A, v)=e(T_3, v) \cdot e(T_1^d \cdot T_2, h)^{-1}$$

(단, $d=\frac{\xi_1}{\xi_2}$)

- 익명자격증명서의 소유자 정보와 자신이 생성한 난수 r을 이용하여 $Q=e(A \cdot u^k, g_2)^r$를 연산 한다.
- 계산 값 $O=e(A, v)$와 자신이 생성한 난수 r'을 이용하여 $R=e(A, v)^{r'}$을 연산한다.
- $S=Q \cdot R$을 연산하고 결과 값을 P와 비교한다.

S와 P가 동일한 값이면, 익명자격증명서는 유효하고, 다른 값이면 익명자격증명서는 유효하지 않은 것으로 판단한다. 검증식의 정확성(correctness)은 아래 수식 (1)을 통해 확인할 수 있다.

$$S = Q \cdot R = e(A \cdot u^k, g_2)^r \cdot e(A, v)^{r'}$$
$$= e(A, g_2^r) \cdot e(u^k, g_2^r) \cdot e(A, v)^{r'}$$
$$= e(A, g_2^r \cdot v^{r'}) \cdot e(u^r, g_2^k) = \mathrm{P} \quad (1)$$

- 익명자격증명서가 유효하다면, 즉 사용자가 요청한 서비스를 이용할 자격을 갖추었음이 증명된다면 서비스제공자는 사용자에게 요청한 서비스를 제공하고, 익명자격증명서가 유효하지 않다면 서비스제공자는 사용자가 요청한 서비스를 거절하고 서비스를 종료한다.

2.4. 익명자격증명서(Qualification Certificate) 구조

익명자격증명서(Qualification Certificate)는 사용자가 자신의 자격을 익명으로 증명하기 위해서 자격증명 발급서버로부터 발급받아서 서비스제공자에게 그룹서명과 함께 제출하는 것이다. 익명자격증명서를 위해서 기존에 알려진 실명 기반 속성 인증서 (PMI)를 수정하여 직접 활용할 수 있다. 이를 위한 익명자격증명서의 구조는 〔표 1〕과 같다.

각 필드의 내용을 살펴보면, 버전정보(Version Number) 필드에는 PMI 인증서(익명자격증명서)의 버전정보를 입력하고, 시리얼 번호(Serial Number) 필드에는 익명자격증명서의 효율적인 관리를 위하여 익명자격증명서를 발급한 발급기관에서 각 익명자격증명서에 부여한 고유한 숫자가 들어간다. 즉, 시리얼 번호 필드와 발급자 필드의 값이 결합하여 유일한 익명자격증명서로 식별이 가능해진다. 익명자격증명서의 내용에 대한 무결성 및 내용의 신뢰성을 확보하기 위해서 발급자의 서명이 추가될 필요가 있는데, 서명에 사용되는 서명 알고리즘의 종류는 서명 알고리즘(Signature Algorithm) 필드에 기술한다. 발급자(Issuer) 필드에는 해당 익명자격증명서를 발급한 발급자의 정보가 들어가고, 유효기간(Validity Period) 필드에는 해당 익명자격증명서의 발급일과 만료일을 기록한다. 이때, 익명자격증명서의 검증을 위해서, 멤버키 발급자(Key Issuer)가 사용자의 그룹 멤버키 A를 이용하여 익명자격증명서에 들어갈 소유자 정보를 생성하는 시점의 정확한 시간을 기록한다. 사용자의 그룹 멤버키 A와 그룹 공개키들 중 g_1, g_2, w는 그룹 멤버의 폐기(revoke)가 있을 때 마다 새로운 값으로 갱신되고, 그룹 서명의 생성과 검증에 멤버키는 물론 그룹 공개키도 이용되기 때문에 어느 시점의 공개키를 이용하여 서명을 생성하고 검증할 지는 중요한 포인트가 된다. 따라서 발급일 정보는 멤버키 발급자(Key Issuer)가 자격증명 발급서버에 정확한 시간을 알려주어 발급일 필드에 넣는 것이 필요하다. 우리는 익명자격증명서를 일회용으로 가정하기 때문에 만료일이 큰 의미가 없을 수 도 있지만, 영어시험 성적 같은 경우 시험 성적의 유효기간 등의 표현에 만료일 값을 이용할 수 있다. 또한 사용자가 자신의 익명성을 훼손하더라도 익명자격증명서를 여러번 사용하고자 한다면 익명자격증명서의 만료일은 더욱 의미를 지니게 될 것이다. 그러나 익명자격증명서를 저장해서 여러번 사용하게 된다면 익명자격증명서 관리의 어려움 및 익명성이 약해지는 등 단점이 생길 수 있으므로 일회용 익명자격증명서의 사용을 권장한다. 소유자(Holder) 필드에는 해당 익명자격증명서의 소유자 정보가 들어가는데, 이 값은 사용자의 익명정보와 익

〔표 1〕 익명자격증명서의 구조

Version Number	PMI 인증서의 버전 정보
Serial Number	PMI 인증서 고유의 시리얼 번호
Signature Algorithm	PMI 인증서 서명에 사용할 서명 알고리즘의 종류
Issuer	인증서 발급자 (ex: ETS 센터)
Validity Period	인증서 유효기간 (발급일, 폐기일)
Holder	**사용자 키와 결합되어 난수화된 사용자 익명정보. 본 제안 기법에서는 $e(A \cdot u^k, g_2)$ 값을 넣음**
Attributes	자격정보 (ex: 토플 ibt 점수 116점)
Issuer Unique Identifier	인증서 발급자에 대한 추가 정보
Extensions	인증서 확장
Signature	인증서에 대한 발급자의 서명

명자격증명서 소유자가 익명자격증명서를 발급받을 때 생성한 난수 값 k로 구성된 값이 입력된다. 즉, 사용자의 그룹 멤버키 A와 사용자가 생성한 랜덤값 k, 그룹 공개키 u, g_2를 이용하여 $e(A \cdot u^k, g_2)$를 연산하여, 이 값을 소유자 필드에 입력한다. 또한 이 값은 자격증명 발급서버가 요청하여 멤버키 발급자(Key Issuer)가 생성하는 값이다. 자격(Attributes) 필드에는 해당 익명자격증명서가 증명하고자 하는 정보가 들어가는데, 예를들면 성인, 회사의 이사진, 영어점수, 장애인 여부, 경로우대 여부 등의 정보가 들어갈 수 있다. Issuer Unique Identifier 필드는 익명자격증명서 발급자에 대한 추가 정보가 기록될 수 있으나 일반적인 PMI 인증서에서 생략하는 경우가 대부분이고, 익명자격증명서에서 꼭 필요한 필드가 아니므로 생략 가능한 필드로 본다. 또한 확장(extension)필드도 옵션으로써, 익명자격증명서에서는 사용하지 않는다. 서명(Signature) 필드는 서명 알고리즘 필드에서 언급된 서명 알고리즘을 사용하여 익명자격증명서 발급자가 [버전정보, 시리얼 번호, 서명 알고리즘, 발급자, 유효기간, 소유자, 속성, Issuer Unique Identifier, 확장필드]에 대해서 보증한다는 뜻으로 위 내용에 대한 서명을 생성하고, 이 서명 값이 포함된다.

2.5. 안전성 분석

본 장에서는 2.2절에서 기술한 익명성 기반의 자격증명 기법이 가져야 할 보안 요구사항들을 상기 제안 기법이 충실히 만족하고 있음을 보인다. 특히, 이전의 알려진 방법들과는 다르게, 인증서 형태의 익명자격증명서를 사용할 때 피할 수 없는 사용자의 행적 연결 문제를 어떻게 해결하였는지에 대해서 중점적으로 설명한다.

(1) 익명자격증명서 도용방지

사용자A가 자신의 익명자격증명서를 사용자B에게 대여해 주거나, 사용자B가 사용자A의 익명자격증명서를 불법으로 취득한 경우를 고려하자. 사용자A의 그룹 멤버키를 A_A라 하고, 사용자B의 그룹 멤버키를 A_B라고 하자. 사용자B가 사용자A의 익명자격증명서, 자신의 그룹서명과 함께 서비스제공자에게 $P = e(A_B, g_2^r \cdot v^{r'}) \cdot e(u^r, g_2^k)$를 전송한다. 서비스제공자는 자신이 알고 있는 값들인 r, r'과 사용자B가 서비스제공자에게 전송한 사용자A의 익명자격증명서의 소유자 정보($e(A_A \cdot u^k, g_2)$), 사용자B가 자신의 그룹멤버키로 생성하여 서비스제공자에게 전송한 그룹서명에서 서비스제공자가 $e(A_B, v)$를 계산할 수 있고, 이를 이용하여 익명자격증명서의 유효성을 검증한다. 즉, $S = e(A_A \cdot u^k, g_2)^r \cdot e(A_B, v)^{r'}$을 계산하고 이 값이 P와 동일한 값인지를 비교한다.

$$\begin{aligned} P &= e(A_B, g_2^r \cdot v^{r'}) \cdot e(u^r, g_2^k) \\ &= e(A_B, g_2^r) \cdot e(A_B, v)^{r'} \cdot e(u^k, g_2^r) \\ &= e(A_B \cdot u^k, g_2^r) \cdot e(A_B, v)^{r'} \end{aligned} \quad (2)$$

식 (2)에서 서비스제공자가 계산한 값인 S와 비교하면, 사용자B가 사용자A의 익명자격증명서를 도용할 경우, 사용자A와 사용자B의 그룹 멤버키 A_A와 A_B는 서로 같을 수가 없으므로 서비스제공자의 익명자격증명서 검증에 통과할 수 없다. 따라서 사용자B는 사용자A의 익명자격증명서를 도용하여 불법으로 서비스제공자가 제공하는 서비스를 이용할 수 없다.

(2) 서비스제공자의 불법행위 방지

서비스제공자가 회원관리 등의 목적으로 사용자의 행동패턴을 링크하기 위해서, 사용자에게 익명자격증명서를 요구할 때 사용자에게 전송하는 값인 $g_2^r \cdot v^{r'}$, u^r 대신 고정된 값 C (단, $C \neq g_2^r \cdot v^{r'}$)와 D (단, $D \neq u^r$)를 전송한다면, 사용자는 $P = e(A, g_2^r \cdot v^{r'}) \cdot e(u^r, g_2^k)$ 대신 $P' = e(A, C) \cdot e(D, g_2^k)$을 계산하여 서비스제공자에게 전송하게 될 것이다. 그러나 사용자가 익명자격증명서 발급을 요청할 때 생성하는 난수 k를 매번 다른 값으로 선택한다면, P값은 매번 다른 값이 생성될 것이므로, 서비스제공자의 의도대로 P'이 동일한 값이 나올 수 없고, 따라서 동일 사용자인지를 알 수 없을 것이다.

(3) 서비스제공자간 공모방지

서비스제공자간 공모가 있다면 사용자의 행적을 추적할 수 있고, 극단적인 경우 사용자에 대한 정보의 조합을 통해서 사용자의 실제 신원을 확인할 수 있을 수도 있다. 이를 막기 위해서 멤버 확인자(Opener)가 서비스제공자마다 다른 비밀키 ξ_1, ξ_2를 이용하여 특정 서비스제공자용 공개키 $u_i^{\xi_1} = v_i^{\xi_2} = h_i$, $1 \leq i \leq m$(단, m은 서비스 제공자의 수)를 생성하여 공개함으로써, 서비스제공자A가 사용자k의 서명 값으로부터 계산하는 값 $e(A_k, v_A)$와, 서비스제공자B가 사용자k의 서명

값으로부터 계산하는 값 $e(A_k, v_B)$는 다른 값을 갖기 때문에 서비스제공자간 공모를 하더라도, 서비스제공자A의 서비스를 제공받은 사용자k와 서비스제공자B의 서비스를 제공받은 사용자k가 동일인인지 여부를 서비스제공자들은 알 수 없으므로 서비스제공자간 공모는 불가능하다. 이에 더해서, 서비스제공자마다 다른 공개키 u, v, h를 이용할 경우 해당 사용자가 어떤 서비스제공자에 접속하여 서비스를 받고자 하는지를 자격증명 발급서버에 드러내지 않기 위하여 서비스제공자에 할당된 공개키 u에 사용자가 생성한 난수 k승을 한 u^k을 사용자가 자격증명 발급서버에 전송하고, 이 값을 이용하여 익명자격증명서가 발급되도록 하였다.

(4) 익명자격증명서의 무결성 및 신뢰성 확보

익명자격증명서가 악의적인 사용자 혹은 공격자에 의해서 수정되지 않았음을 증명(무결성)하고, 익명자격증명서의 발급자 필드에 적혀있는 발급자가 발급한 것이 맞음을 확인(신뢰성)할 수 있도록 하기 위해서 익명자격증명서의 마지막에 익명자격증명서 발급자의 서명이 들어간다. 따라서 익명자격증명서의 서명을 검증함으로써 익명자격증명서에 대한 무결성 및 신뢰성을 확인할 수 있다.

(5) 자격증명서 비연결성 (Certificate-unlinkability)

익명자격증명서를 이용하여 접근제어를 할 때, 익명자격증명서 때문에 사용자의 인터넷 상에서의 행적이 링크되어, 사용자 프라이버시가 침해되어서는 안 될 것이다. 본 논문에서는 이를 위하여 익명자격증명서의 소유자 필드의 값이 매번 바뀌도록 하였다. 즉, 그룹 멤버키 A값이 그룹 멤버들 중 한 사람이 폐기(revoke) 될 때 마다 바뀌는 값이긴 하지만 일정 시간동안(사용자가 익명자격증명서를 여러 번 발급받아 사용하는 동안) 그룹 멤버의 폐기(revoke)가 한 번도 일어나지 않는다면, 익명자격증명서의 소유자 정보가 동일한 값을 유지해서 사용자의 행적이 링크되는 것을 막기 위하여 사용자는 익명자격증명서를 발급받을 때 마다 사용자만이 알고 있는 난수 k를 새롭게 생성하여 익명자격증명서 소유자 정보에 반영될 수 있도록 하였다.

2.6. 성능 분석

제안한 익명성 기반의 자격증명 기법은 SGS-LL을 이용한 그룹서명 기법에 자격증명을 위한 자격증명서 발급 및 검증과정이 추가된 것이다. 익명성을 제공하는 자격증명을 위해서 추가된 연산은 사용자의 입장에서 1회의 지수 연산과 2번의 페어링 연산이 추가되고, 서비스 제공자의 입장에서는 자격증명서 요청시 3회의 지수연산이 추가되고(이 연산은 미리 연산을 해둘 수 있기 때문에 실시간 연산의 수에는 포함되지 않음), 자격증명서 검증시 2회의 페어링 연산과 3회의 지수연산이 추가된다. 또한 멤버키 발급자가 익명자격증명서의 소유자 필드에 포함될 값을 연산하는데 1회의 지수연산과 1회의 페어링 연산이 추가된다. 익명성을 제공하는 접근제어에 제안한 기법을 적용하기 위해서 추가되는 연산의 횟수는 〔표 2〕에 정리하였다. 본 절의 설명과 〔표 2〕에서 알 수 있듯이 본 논문에서 제안한 기법을 활용하여 사용자의 신원정보를 드러내지 않고, 안전하고 편리하게 인터넷을 이용하는데 추가되는 연산 모듈은 아주 적은 양이다. 또한 이들 연산은 성능이 좋은 각 서버와 사용자 PC에서 수행되기 때문에 연산의 추가로 인한 서비스 딜레이는 사용자가 느끼기에 아주 미미하리라 생각된다.

2.7. 향후 과제

본 논문에서 제안한 기법을 서비스에 적용할 때 사용자의 편의성을 높이기 위해서, 서비스에 대한 접근제어가 필요한 순간마다 사용자가 익명자격증명서를 발급받아서 서비스제공자에 전송하지 않고, 사용자가 한 번 익명자격증명서를 서비스제공자에 등록하고, 그

〔표 2〕 제안한 기법으로 접근제어를 할 경우 추가되는 연산의 수

	지수연산 횟수	페어링연산 횟수	연산시점
멤버키 발급자	1	1	익명자격증명서 소유자 필드 값 연산시
사용자	1	2	서비스 이용시
서비스 제공자	3(0)	0	익명자격증명서 요청시
	3	2	익명자격증명서 검증시

기록을 유지할 수 있는 구조가 된다면 더욱 편리한 서비스 이용이 가능할 것이다. 제안한 익명성 기반의 속성 기반 접근제어 기법에서, 사용자가 익명자격증명서를 서비스제공자에 전송하고, 서비스제공자가 이를 검증할 때, 사용자의 그룹서명에서 $O=e(A,v)$ 값을 계산해 낸다. 이 값은 사용자의 그룹 멤버키 쌍의 일부인 A 값이 변하지 않는 한 동일한 값으로 유지되는 값이기 때문에 사용자가 특정 서비스에 접근하고자 할 때마다 동일한 자격증명서를 매번 발급받아서 서비스제공자에 제시할 필요가 없을 수 있다. 그러나 사용자의 그룹 멤버키 A 값은 그룹서명의 특성상, 그룹 멤버 중 한 사람이라도 그룹 멤버키가 폐기되면 업데이트되어 수정되는 값이므로 $O=e(A,v)$가 동일한 값을 유지하는 기간이 길지 않다. 이는 사용자 편의성 측면에서는 장점이 될 수 있지만, 사용자의 완벽한 익명성 제공측면에서는 짧은 기간이지만 사용자의 행적이 연결(linkable) 될 수 있다는 점을 고려할 때 단점이 될 수도 있다. 즉, 하나의 서비스제공자 내에서 사용자의 행적이 연결(linkable)되어 사용자의 완벽한 익명성을 해칠 수 있다는 점과, 사용자의 편의성 사이에는 트레이드 오프(trade off)가 있어야 할 것으로 보인다.

또한, 제안한 기법에서 자격증명 발급서버는 익명도메인에 포함된 기관이 아니라 실명도메인에 포함된 기관으로써, 사용자의 실제 신분정보를 알고 있고, 사용자들의 특정 익명자격증명서를 관리하고 발급하는 기관이다. 즉, 자격증명 발급서버는 사용자의 재직 증명서를 발급할 수 있는 회사일 수도 있고, 공인 토플점수에 대한 증명서를 발급해 주는 토플위원회가 될 수도 있다. 또한 의사자격증명서를 발급해 주는 의사협회가 될 수도 있다. 따라서 자격증명 발급서버는 개인의 신분정보와 자격정보를 모두 알고 있는 공신력 있는 기관인 대신, 사용자의 익명 멤버키 정보에 대해서는 전혀 알 수 없다. 반면에, 서비스제공자는 사용자의 신분정보는 전혀 알지 못하지만, 사용자가 발급받아온 익명자격증명서들을 볼 수 있는 기관이므로 해당 익명자격증명서를 발급한 자격증명 발급서버와 공모하게 된다면, 서비스제공자가 해당 익명 사용자의 신분을 알 수 있다는 단점이 있다. 이러한 공모 문제에 대한 해결방안은 향후 흥미로운 연구 주제가 될 것이다.

III. 제안 기법을 적용한 익명 지불 방법

온라인상의 지불 시스템에서 사용자에게 익명성을 제공하기 위한 하나의 방안으로 본 논문에서 제안한 익명성을 제공하는 자격 증명 기법을 적용할 수 있다. 즉, 익명자격증명서를 발급하는 주체를 은행 혹은 온라인 상품권 등의 판매처로 하고, 익명자격증명서의 자격(attribute) 필드에 금액을 기록하여 익명자격증명서를 지불토큰으로 이용할 수 있을 것이다. 즉, 사용자는 서비스제공자 서버에 접속하여 자신이 원하는 물건의 가격을 알고, 이 가격에 맞는 지불토큰을 은행 혹은 온라인 상품권 판매처 등에 돈을 지불한 후 발급받아서 서비스제공자에 전송하여 이용하면 된다. 그리고 지불토큰의 유효성 검증은 익명자격증명서의 유효성 검증과 동일한 방법으로 할 수 있다.

익명자격증명서를 지불토큰으로 사용할 경우, 지불토큰은 반드시 일회용으로 사용되어야 한다. 이는 서비스제공자가 지불토큰 검증시 해당 지불토큰의 중복사용을 체크할 수 있으므로 (지불토큰은 서비스제공자마다 다른 공개키 u,v,h를 이용하여 생성되므로, 특정 서비스제공자만이 검증할 수 있고, 따라서 해당 서비스제공자는 사용자가 해당 지불토큰을 중복 사용했는지 여부를 알 수 있다.) 문제가 되지 않는다.

IV. 결론

인터넷이 우리에게 가져다주는 편리함과 빠른 일처리의 이면에 개인정보의 유출 및 프라이버시 침해의 정도가 심각한 수준에 이르렀다. 웹사이트에서 유출된 개인정보는 실제 돈을 받고 거래되기도 하고, 제2의 범죄에 악용되기도 한다고 한다. 이를 막기 위한 하나의 대안으로 익명인증을 생각할 수 있으며, 익명인증을 제공하는 하나의 방법으로써 본 논문에서는 SGS-LL을 이용하였고, 이는 [7]에 Link 알고리즘을 추가한 것이다. [7]은 그룹 멤버의 수와 관계없이 일정한 길이의 서명을 생성하고, 1024비트 RSA와 동등한 암호학적 안전성을 제공하면서 비교적 짧은 길이의 서명으로 사용자의 익명인증기능을 제공한다는 장점이 있다. 그러나, 익명인증만으로는 웹서비스 제공에 한계가 있으며, 이를 해결하기 위해서 본 논문에서는 익명자격증명서를 SGS-LL과 함께 이용하여 웹서비스 제공시 접근제어가 가능하도록 하여 사용자 맞춤형 서비스 제공이 가능하도록 하였다. 본 논문에서 제시한 익명성을 제공하는 속성 기반 접근제어 기법을 이용하면, 사용자의 익명성을 보장하면서도, 웹 서비스제공자가 RBAC(Role Based Access Control)처럼 유연한 접근제어를 할 수 있다는 장점이 있다.

즉, A회사 사람들만 사용할 수 있는 서비스의 경우, A회사 사원임을 증명하는 익명자격증명서를 가진 사람, A회사 임원 혹은 팀장임을 증명하는 익명자격증명서를 가진 사람들 모두 접근할 수 있도록 접근제어를 설정할 수 있고, 장애인들만 사용할 수 있는 서비스의 경우, 장애인임을 증명하는 익명자격증명서를 가진 사람뿐만 아니라 장애3등급의 익명자격증명서를 가진 사람도 접근할 수 있도록 접근제어를 설정할 수 있다. 또한 해당 익명자격증명서를 소유한 사람이 이용 가능한 다른 서비스도 이용할 수 있도록 설정할 수 있을 것이다. 즉, 'ㄱ' 익명자격증명서를 소유한 사람은 A, B, C 서비스를 이용할 수 있고, 'ㄴ' 익명자격증명서를 소유한 사람은 A, C, D 서비스를 이용할 수 있도록 설정하여 접근제어를 하는 RBAC(Role Based Access Control)기법의 적용이 가능하다. 그러나, 2.7절에서 기술하였듯이 본 논문에서 제안한 방법은, 동일한 사이트에서 서비스를 두 번 이상 이용한 경우 그룹 멤버 중 한 사람이라도 키 갱신 혹은 멤버 탈퇴가 없었다면 동일 서비스제공자 내에서 사용자의 행적이 연결될 수 있는데, 이는 익명 서비스를 이용하는데 있어서 사용자에게 편리함을 줄 수 있지만, 단점이 될 수도 있을 것이다. 이 문제에 대한 해결 방안은 향후 연구주제로 남겨둔다.

참고문헌

[1] Benjumea, V.Lopez, J.Montenegro, J.A. Troya, and J.M. "A first approach to provide anonymity in attributes certificates," *PKC 2004, LNCS 2947*, pp. 402-415. 2004.

[2] A. Kiayias, Y. Tsiounis, and M. Yung, "Traceable Signatures," *Eurocrypt 2004, LNCS 3027*, pp.571-589, 2004.

[3] D.Chaum and E. van Heyst, "Group signatures," *Proceedings of Eurocrypt 1991, LNCS 547*, pp. 257-265, 1991.

[4] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups," *Proceedings of Crypto 1997, LNCS 1296*, 1997.

[5] G. Ateniese, J. Camenisch, M. Joye and G. Tsudik, "A practical and provably secure coalition-resistant group signature scheme," Proceedings of Crypt 2000, LNCS 1880, pp.255-270, 2000.

[6] L. Nguyen and R. Safavi-Naini, "Efficient and probably secure trapdoor-free group signature schemes from bilinear pairings," Proceedings of Asiacrypt 2004, LNCS 3329, pp. 372-386, 2004.

[7] D. Boneh, X. Boyen and H. Shacham, "Short group signatures," Proceedings of Crypto 2004, LNCS 3152, pp. 41-55, 2004.

[8] D. Boneh and X. Boyen. "Short signatures without random oracles," Proceedings of Eurocrypt 2004, LNCS 3027, pp.56-73, 2004.

[9] S. Kiyomoto, K. Fukushima, and T. Tanaka, "Design of anonymous attribute authentication mechanism," IEICE Trans. Commun., Vol. E92-B, no.4, pp.1112-1118, April 2009.

[10] 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.

[11] 신수연, 권태경, "그룹 서명 기반 익명 인증 및 권한 검증에 관한 연구," 한국정보보호학회 하계학술대회 논문집, 19(1), pp. 357-361, 2009년 7월.

[12] 신수연, 권태경, "프라이버시 보호를 위한 익명 인가에 관한 연구," 한국정보보호학회 동계학술대회 논문집, 19(2), pp. 361-367, 2009년 12월.

[13] Michael Backes, Jan Camenisch, and Dieter Sommer, "Anonymous yet Accountable Access Control," WPES 2005, proceeding of the 2005 ACM workshop on Privacy in the electronic society, pp. 40-46, Nov. 2005.

[14] K. Ren, W. Lou, K.Kim, and R.Deng, "A novel privacy preserving authentication and access control scheme for pervasive computing environments," IEEE Transaction on Vehicular Technology, pp. 1373-1384, July, 2006.

[15] Chuna-Ta Li, Min-Shiang-Hwant, and Yen-Ping Chu, "Further improvement on

a novel privacy preserving authentication and access control scheme for pervasive computing environments," The International Journal for the Computer and Telecommunications Industry, pp. 4255-4258, Dec. 2008.

[16] Kilho Shin and Hiroshi Yasuda, "Provably Secure Anonymous Access Control for Heterogeneous Trusts," ARES 2006, pp. 24-33, April, 2006.

[17] Kilho Shin and Hiroshi Yasuda, "Practical Anonymous Access Control Protocols for Ubiquitous Computing," Journal of Computers, pp. 1-12, Dec. 2006.

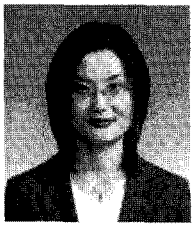
[18] K. Shin. Digital Qualification: An approach to infrastructures of access control for internet commerce," SSGRR 2001, Aug. 2001.

[19] N. Li and J. C. Mitchell. "RT:A Role-based Trust-management Framework," In DARPA Information Survivability Conference and Exposition, pp. 201-212, April, 2003.

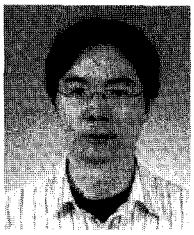
[20] 강전일, 양대헌, 이석준, 이경희, "실생활 응용을 위한 짧은 그룹 서명 기법(BBS04)에 대한 연구," 정보보호학회논문지 19(5), pp.3-15, 2009년 10월.

[21] R. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," Proceeding of Asiacrypt 2001, LNCS2248, pp.552-565, Dec. 2001.

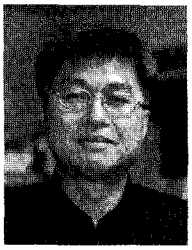
〈著者紹介〉



이 윤 경 (Yun-kyung Lee) 정회원
1999년 2월: 경북대학교 전자전기공학부 졸업
2001년 2월: 포항공과대학교 전자공학과 석사
2001년 1월~현재: 한국전자통신연구원 선임연구원
〈관심분야〉 인증 및 인가 메커니즘, 암호 프로토콜, 스마트폰 보안 등



황 정 연 (Jung Yeon Hwang) 정회원
1999년 2월: 고려대학교 수학과 졸업
2003년 2월: 고려대학교 정보보호대학원 공학석사
2006년 8월: 고려대학교 정보보호대학원 공학박사
2009년 5월: 고려대학교 BK21 유비쿼터스정보보호사업단 연구교수
2009년 5월~현재: 한국전자통신연구원 선임연구원
〈관심분야〉 암호프로토콜, 정보보호이론, 프라이버시강화기술(PET) 등



정 병 호 (Chung, Byung Ho) 정회원
1988년 2월: 전남대학교 전산통계학과 졸업
2000년 2월: 충남대학교 컴퓨터과학과 석사
2005년 8월: 충남대학교 컴퓨터과학과 박사
〈관심분야〉 정보보호, 무선통신프로토콜

김 정 녀 (Jeong Nyeo Kim) 종신회원
1987년 2월: 전남대학교 전산통계학과 졸업
2000년 2월: 충남대학교 컴퓨터공학과 석사
2004년 2월: 충남대학교 컴퓨터공학과 박사
1996년: OSF/RI 공동연구 파견(미국)
2005년: Univ. of California, Irvine Post-Doc.
1988년~현재: 한국전자통신연구원 휴먼인식기술연구팀장 책임연구원
〈관심분야〉 시스템·네트워크보안, 보안 OS, 바이오정보보호 등