그룹서명을 이용한 익명 인증 구조

이윤경* · 정병호*· 김정녀*

*한국전자통신연구원

Anonymous Authentication Framework usign Group Signature

Yun-kyung Lee* · Byung-ho Chung*· Jeong-nyeo Kim*

*Electronics and Telecommunications Research Institute

E-mail : neohappy@etri.re.kr

요 약

인터넷 이용이 활성화 되면서 각종 웹 서버에서의 개인정보 과다 수집 및 노출이 큰 이슈가 되고 있다. 인터넷이 우리 생활에 주는 편리함을 그대로 누리면서 개인정보를 보호할 수 있는 방안으로 익명인증 기술이 있을 수 있다. 익명인증은 익명성을 제공하는 디지털서명을 이용한 인증 방법이다. 익명성을 제공하는 디지털서명 방법은 전자화폐와 전자투표 시스템 등의 응용을 위해서 주로 연구되어 왔으나, 최근에는 인터넷 환경에서 개인정보 보호를 위한 익명인증 방법의 하나로써 연구되고 있다. 본 논문에서는 익명성을 제공하는 전자서명 방법 중 그룹서명 방식에 관하여 소개하고, 그룹서명을 이용한 익명인증 시스템의 구조를 제안하고자 한다.

키워드

Anonymous Authentication, Group Signature, Privacy

Ⅰ. 서 론

인터넷이 활성화됨에 따라 인터넷 이용의 순기능도 많지만 역기능도 드러나고 있는 실정이다. 대표적인 역기능으로 불필요한 개인정보의 노출을 꼽을 수 있다. 인터넷의 특성상 개인정보 수집 주체의 실수 등으로 수집된 개인정보가 한 번 노출되면 삽시간에 너무 많은 사람들이 그 정보를 접할 수 있다는 사실은 더욱 큰 문제가 된다. 그리고 일상생활에서 프라이버시는 새로운 것이 아니다. 사람들이 상점에서 어떤 것을 구입할 때 어느 누구도 구매자의 신원을 물어보지 않는다. 또한 우체국에서 편지를 보낼 때에도 편지 보내는 사람이 누구인지 물어보지 않고, 반송 주소가 정확한지를 체크하지도 않는다. 그러나 인터넷에서는 대부분의 경우 개인의 신분 확인 과정이 필요하고, 암호화 하지 않고 보낸 메일들은 e-mail 시스템에 대한 약간의 지식이 있는 사람이라면 누구든 다른 사람의 메일을 읽을 수 있고, 수정하여 재전송 할 수도 있다[8]. 이러한 상황에 대해서 많은 인터넷 이용자들이 공감하고, 자신의 개인정보 보호에대한 인식이 확대되고 있으며, 더 나아가서 개인정보 보호 및 프라이버시 보호에 대한 요구도 높아지고 있다. 그리고 인터넷 이용자들은 본인이 인터넷으로 하는 모든 행동들이 수집되고 기록될 수 있다는 사실에 의해 야기되는 문제에 대해 특별한 관심을 보이고 있고, 따라서 익명성의 필요성을 공감하고 있다[7].

본 논문에서는 이러한 요구에 발맞추어 개인정보의 노출을 최소화 하면서, 기존의 인터넷 서비스의 많은 부분을 그대로 이용할 수 있는 익명인증 시스템의 구조에 대하여 제안하고자 한다. 이 논문의 2장에서는 익명성을 제공할 수 있는 여러 가지 서명 기법들 중 본 논문에서 이용할 그룹서명 기법의 개념을 설명하고, 3장에서는 D.Boneh가 제안한 그룹 서명 방법을 간략히 살펴본 후, 4장에서 D. Boneh의 그룹서명 방법을 이용한 익명 인증 시스템에 관하여 기술하고자 한다.

Ⅱ. 그룹서명(Group Signature)

그룹 서명은 특정 그룹의 멤버임을 인증하는 방법을 제시한다. 즉, 그룹 서명을 검증하는 측에서는 그 서명을 생성한 사람이 그 그룹의 멤버임을 알 수 있지만, 그룹 멤버 중 누구인지는 알 수 없기 때문에 익명성(anonymity)을 제공할 수 있

다. 그룹 서명의 기본 개념은 1991년 David Chaum과 Eugene van Heyst에 의해서 처음 t hro되었다[1]. [1]에 따르면, 그룹 서명은 다음 세 가지 특성을 가져야 한다.

(1) 그룹 멤버만이 메시지에 서명을 할 수 있다.

(2) 서명을 받은 사람은 유효한 그룹 서명인지 확인할 수는 있지만, 어떤 그룹 멤버가 그 서명을 생성했는지는 알 수 없다.

(3) 필요하다면, 메시지에 서명한 사람이 누구인지를 밝힐 수 있다.

그룹 서명에서 가장 중요한 기능을 하는 그룹 매니저는 그룹 멤버를 추가하고, 논쟁이 발생한 경우 서명한 사람의 신원을 공개할 수 있는 권한을 갖는다. 다양한 그룹 서명 방법이 제안되고 있지만, 대부분의 그룹 서명 방법에서는 다음의 기본 구조를 따른다.

(1) Key generation : 몇 개의 시큐리티 파라미터들을 이용하여 그룹 공개키와 그룹 멤버들의 비밀키를 생성하는 과정.

(2) Join : 그룹 매니저와 사용자 사이의 프로토콜로써, 그룹의 멤버가 되고자 하는 사용자가 그룹 매니저로부터 멤버쉽 인증서와 멤버쉽 비밀키를 받는 과정.

(3) Sign : 그룹 멤버가 그룹의 명의로 메시지에 서명하는 과정.

(4) Verify : 그룹 공개키로 서명을 검증하는 과정.

(5) Open : 논란이 발생할 경우, 그룹 매니저가 그룹 비밀키와 서명을 이용하여 서명한 사람의 신원을 밝히는 과정.

위 5가지 과정 외에 최근에 발표된 그룹 서명 관련 논문에는 "Revocation" 과정을 추가로 소개하기도 한다. Revocation이란 그룹 멤버 자격을 박탈하는 과정을 의미하는데, revocation 방법은 그룹 서명의 구조에 따라서 달라진다. 지금까지 제안된 revocation 방법에는 아래와 같이 크게 세 가지가 있다[5].

(1) Revoke 하고자 하는 사용자를 제외한 모든 사용자에게 새로운 서명 확인 키를 전달하는 방법으로, 비밀키는 서명자(signer) 개개인에게 따로 전달하고, 공개키는 모든 서명자와 서명 확인자(verifier)에게 브로드캐스트 메시지로 전송한다.

(2) 모든 서명자와 서명 확인자에게 브로드캐스트 메시지를 전송한다[2].

(3) Revocation 메시지를 서명 확인자에게만 전송한다[3,4,5].

그룹 서명에서 요구하는 시큐리티로는 correctness, unforgeability, anonymity, unlinkability, exculpability 등이 있는데, correctness, unforgeability, anonymity는 그룹 서명에서 반드시 제공해야 할 기본적인 시큐리티 요소이고, unlinkability, exculpability 등은 그룹 서명의 필수요소는 아니지만, 제공되면 더욱 강력한 익명인증 방법을 제공할 수 있는 시큐리티 요소이다. 각 시큐리티 요소의 의미는 다음과 같다.

(1) correctness : 그룹 멤버들의 유효한 서명은 항상 제대로 검증되어야 하고, 유효하지 않은 서명은 반드시 서명 검증에 실패하여야 한다.

(2) unforgeability : 그룹의 멤버들만이 유효한 서명을 생성할 수 있어야 한다.

(3) anonymity : 메시지와 그 메시지에 대한 서명이 주어졌을 때, 그룹 매니저 이외의 사람은 서명자의 신분을 알 수 없어야 한다.

(4) unlinkability : 두 메시지와 각 메시지에 대한 서명이 주어졌을 때, 그 서명들이 동일한 서명자가 서명한 것인지 여부를 알 수 없어야 한다.

(5) exculpability : 모든 다른 그룹 멤버들과 매니저가 결탁하더라도, 참여하지 않은 그룹 멤버에 대한 서명을 위조할 수 없어야 한다.

III. Short Group Signatures[6]

지금까지 제안된 대부분의 그룹서명 기법은 그룹 멤버의 수에 비례하여 서명의 길이가 증가한다는 단점이 있었다. 그러나 D. Boneh가 2004년에 발표한 논문[6]에서 제시한 그룹서명 방법에서는 그룹 멤버의 수와 관계없이 일정한 길이의 서명값을 생성할 수 있고, 서명의 길이도 RSA 서명과 비슷한 길이인 1533비트 이다. 또한 이 논문에서는 그룹 멤버를 revoke할 수 있는 방법도 제시하고 있다.

Short Group Signature(SGS)는 q-strong Diffie-Hellmann 가정에 그 안전성의 기반을 두고, linear encryption 방법을 적용한 그룹서명 구조를 제시하고 있다. Q-strong Diffie-HEllmann 가정이란 두 개의 순환군(crclic group) G1, G2 이고, G1의 생성자(generator) g1, G2의 생성자 g2가 있을 때, e : $G_1 X G_2 \rightarrow G_T$인 bilinear maps에 대해서 () 의 입력이 주어졌을 때, (, x)를 구하기 어렵다는 것을 전제로 하는 가정이다.

SGS는 KeyGen, Sign, Verify, Open, Revoke의 다섯 단계로 구성되어 있다.

KeyGen(n) : n명의 그룹 멤버에 대해서 그룹 공개키, 그룹 멤버들의 비밀키, 그룹 매니저의 비밀키를 생성하는 과정으로, 생성 방법은 다음과 같다.

(1) G1에서 h를 랜덤으로 선택하고, Z_p^*에서 ξ1,ξ2를 랜덤으로 선택한다. 이때 (ξ1,ξ2)는 그룹 매니저의 비밀키 gmsk로 한다.

(2) 를 만족하는 u, v를 G_1에서 선택한다.

(3) Z_p^*에서 ϒ를 임의로 선택하여 $w=gu^{\xi_1} = v^{\xi_2} = h_2^{\gamma}$인 w를 생성한다.

(4) ϒ를 이용하여 $1 \le i \le n$ 인 각 사용자 i에 대해서 SDH 집합 (A_i, x_i)를 생성하여 각 사용자의 비밀키로 한다.

Z_p^*에서 xi를 랜덤으로 선택하고, 를 Ai로 한다.

(5) 그룹 공개키 gpk를 (g1, g2, h, u, v, w)로 한다.

Sign(gpk, gsk[i], M) : 그룹 공개키 gpk = (g1, g2,

h, u, v, w)와 사용자의 비밀키 gsk[i] = (A_i,x_i),서명할 메시지 $M \in \{0,1\}^*$에 대해서 서명값 $\sigma = (T_1, T_2, T_3, c, S_\alpha, S_\beta, S_\chi, S_{\delta_1}, S_{\delta_2})$를 생성하는 과정으로, 서명 과정은 다음과 같다.

(1) Z_p 의 원소 중 α, β를 임의로 선택하고, 선형 암호화(linear encryption)를 수행한다.

(2)$\delta_1 \leftarrow x\alpha, \delta_2 \leftarrow x\beta$ 인 δ_1과 δ_2를연산한다.

(3) Z_p 의 원소 중 r_α, r_β, r_γ, r_{δ_1}, r_{δ_2}을 임의로 선택하고, R_1, R_2, R_3, R_4, R_5를 다음의 수식에 따라 연산한다.

$$R_1 \leftarrow u^{r_\alpha}$$
$$R_2 \leftarrow v^{r_\beta}$$
$$R_3 \leftarrow e(T_3,g_2)^{r_x} \cdot e(h,w)^{-r_\alpha - r_\beta} \cdot e(h,g_2)^{-r_{\delta_1} - r_{\delta_2}}$$
$$R_4 \leftarrow T_1^{r_x} \cdot u^{-r_{\delta_1}}$$
$$R_5 \leftarrow T_2^{r_x} \cdot v^{-r_{\delta_2}}$$

(4) 서명할 메시지 M과 앞서 연산한 T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5를 랜덤오라클의 입력으로 하여 결과값 c를 생성한다.

$$c \leftarrow H(M, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5) \in Z_p$$

(5) c를 다음 수식에 적용하여 값을 연산한다.

$$s_\alpha = r_\alpha + c\alpha$$
$$s_\beta = r_\beta + c\beta$$
$$s_x = r_x + cx$$
$$s_{\delta_1} = r_{\delta_1} + c\delta_1$$
$$s_{\delta_2} = r_{\delta_2} + c\delta_2$$

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

Verify(gpk, M, σ) : 그룹 공개키 gpk = (g1, g2, h, u, v, w)와 메시지 M, 그룹 서명 값 σ를 이용하여 서명 값이 유효한지를 검증하는 과정으로, 서명 값이 아래의 5개 수식을 만족하면 유효한 서명이고, 그렇지 않으면 그 서명은 유효하지 않은 서명이다.

$$u^{s_\alpha} \stackrel{?}{=} T_1^c \cdot R_1$$
$$v^{s_\beta} \stackrel{?}{=} T_2^c \cdot R_2$$
$$e(T_3,g_2)^{s_x} \cdot e(h,w)^{-s_\alpha - s_\beta} \cdot e(h,g_2)^{-s_{\delta_1} - s_{\delta_2}} \stackrel{?}{=} \left(e(g_1,g_2)/e(T_3,w)\right)^c \cdot R_3$$
$$T_1^{s_x} u^{-s_{\delta_1}} \stackrel{?}{=} R_4$$
$$T_2^{s_x} v^{-s_{\delta_2}} \stackrel{?}{=} R_5$$

Open(gpk, gmsk, M, σ) : 그룹 공개키 gpk = (g1, g2, h, u, v, w)와 그룹 매니저의 비밀키 gmsk =(ξ_1,ξ_2),메시지 M, 메시지 M에 대한 서명 값 $\sigma = (T_1, T_2, T_3, c, S_\alpha, S_\beta, S_\chi, S_{\delta_1}, S_{\delta_2})$를 이용하여 서명 값 σ를 생성한 그룹 멤버를 찾는 과정으로, 다음과 같다.

(1) 서명 값 σ가 메시지 M에 대해 유효한 서명인지 확인해 본다.

(2) T_1,T_2,T_3와 아래 수식을 이용하여 사용자의 비밀키 값인 A 값을 찾아낸다.

Revoke : Revocation Authority(RA)가 revoke할 그룹 멤버들의 비밀키들로 구성된 Revocation List(RL)를 생성해서 공개하면 모든 서명자와 서명 확인자가 RL을 알게 되고, 이를 통해서 그룹 멤버들은 자신의 공개키를 업데이트하고, 서명 검증에 이를 적용함으로써 revoke된 사용자가 생성한 서명인지 여부를 알 수 있다.

D. Boneh의 SGS는 정직하게 생성된 모든 서명을 검증할 수 있고, 서명 생성자를 정확하게 추적할 수 있으며(correctness), 서명 만으로는 서명 생성자를 알 수 없고(full-anonymity), 그룹 매니저나 여러명의 그룹 멤버들의 공모에 의해 생성된 서명일 지라도, 서명 생성자를 추적할 수 있는 기능을 제공한다(full-traceability). 이러한 특성을 인증 시스템에 적용함으로써 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 없는 특성을 이용함으로써, 특정 자격을 가진 정당한 사용자임을 확인할 수 있는 익명 인증 메커니즘에 D. Boneh의 SGS를 적용할 수 있으리라 본다. 또한 정당한 사용자가 그룹 내에서 허용되지 않는 행동을 했을 경우, 그룹 매니저가 서명한 사람의 신분을 밝힐 수 있도록 함으로써 비상식적인 행동을 한 사람을 추적할 수 있는 방법을 제공한다.

IV. 익명인증 시스템

본 장에서는 3장에서 기술한 D. Boneh의 SGS를 이용한 익명인증 시스템에 관하여 기술하고자 한다. 본 논문에서 제안하는 익명인증 시스템은 실명을 확인하는 과정, 실명 확인 후 개인의 비밀키를 발급받는 과정, 개인의 비밀키로 메시지에 대한 서명을 하고, 이 서명 값으로 인증을 받는 과정, 인증 후 서비스 제공자로부터 서비스를 제공 받는 과정으로 구성되어 있다. 그림 1에서 이들 과정을 간략하게 보여주고 있다.

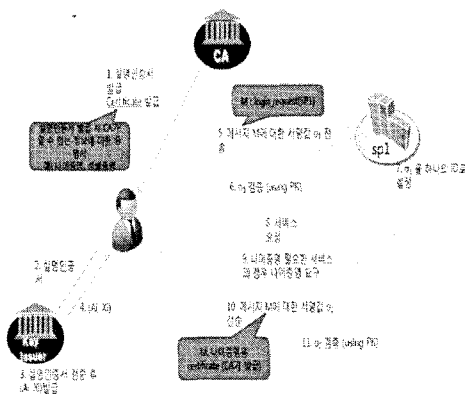

그림 1. 익명인증 시스템 구성

익명 인증을 통하여 인터넷을 이용하더라도 추후 필요한 경우 실명 추적을 할 수 있어야 하기 때문에 실명확인을 거친 후 익명 인증에 사용할 비밀키를 부여한다. 실명확인 방법으로는 여러 가지가 있을 수 있으나, 널리 사용되고 있는 인증서를 이용하는 방법과, 직접 대면에 의한 방법이 있을 수 있다. 인증서를 이용하는 경우, 사용자는 본인의 인증서를 이용하여 key issuer에게 실명인증을 받고, key issuer는 실명을 확인한 후 사용자의 실명과 익명 인증에 사용할 비밀키에 대한 연관 관계를 안전한 데이터베이스에 저장해 둔다. 그리고 사용자의 비밀키(Ai, Xi)를 사용자에게 전송한다. 사용자는 자신의 비밀키를 이용하여 서명값을 생성하고, 생성된 서명 값과 서명한 메시지를 서비스 제공자(SP1)에게 전송하면 SP1은 서명을 확인한 후, 유효한 서명이면 사용자에게 서비스를 제공할 수 있다. 사용자가 SP1에 로그인 할 때, "login_request(SP1)" 이라는 메시지에 대해서 서명을 생성하고, 그 서명 값을 SP1에 전송하는 방법을 사용한다. 또한 SP1은 그 서명 값을 하나의 ID로 생각함으로써 한 사용자의 특성에 맞는 서비스를 지속적으로 제공할 수 있게 된다. 이러한 로그인 방법을 이용함으로써 사용자는 자신의 개인신분을 SP1에게 드러낼 필요가 없으므로 프라이버시 보호 효과를 얻을 수 있고, 사이트마다 사용할 아이디와 패스워드를 따로 기억할 필요가 없으므로 편리함을 누릴 수 있다. 또한 서비스 제공 서버는 개인정보 보호를 위해 필요로하는 노력과 비용을 줄일 수 있고, 사용자가 익명이긴 하지만 동일한 ID로 (즉, 동일한 서명 값으로) 해당 서버에 접속해 오기 때문에 그 사용자의 성향을 파악할 수 있으며 이러한 정보를 실명인증 때와 마찬가지로 마케팅에 이용할 수 있다는 장점이 있다. 다만, 서비스 제공자가 유지해야 하는 사용자의 ID의 길이가 길어진다는 단점이 있으나, 이는 컴퓨터를 이용하는 시스템에서는 큰 문제가 아니라고 보여진다. 사용자마다 다른 비밀키를 가지고 있으므로 동일한 로그인 메시지 "login_request(서비스제공자명)"에 대한 서명 값은 사용자마다 달라지므로 이러한 서명 값을 사용자의 ID로 이용할 수 있다. 그러나 SGS에서 동일 메시지에 대해서 동일한 사용자가 매번 동일한 서명 값을 생성하기 위해서는 랜덤 값인 α, β, 의값이고정된값이어야한다. 이들 랜덤 값을 고정된 값으로 사용할 경우 그룹 서명 기법의 안전성을 약화시킬 수 있으나 일정 주기마다 이들 값들을 업데이트 한다면 어느 정도의 안전성은 보장되리라 본다.

V. 결 론

본 논문에서는 익명성을 제공하는 여러 가지 서명 기법 중 그룹서명 방법을 이용한 익명인증 시스템의 구조를 제안 하였다. 본 논문에서 제안한 익명인증 방법에 따르면, 인터넷 이용자는 개인정보를 서비스제공자에게 일일이 제공할 필요가 없고, 서비스 제공자는 개인정보를 관리해야 하는 부담에서 벗어날 수 있으면서, 마케팅에 이용할 수 있도록 사용자가 일정한 아이디로 접속할 수 있는 방법을 제시하고 있다. 또한 사용자가 특정 서비스를 제공받는데 있어서 사용자에 관한 정보가 필요하다면 믿을만한 기관에서 발급받은 증명서를 서비스제공자에게 제출함으로써 서비스를 제공받을 수 있다. 사용자가 타인에게 피해를 입히거나 불법적인 행위를 할 경우 익명인증을 받은 사용자의 신원을 확인할 수 있는 방법도 제시한다.

참고문헌

[1] D. Chaum and E. van Heyst(1991). "Group Signatures," EUROCRYPT 91', LNCS547, pp.257-265, 1991.

[2] J. Camenisch and A. Lysyanskaya. "Dynamic accumulators and application to efficient revocation of anonymous credentials," CRYPTO 2002, LNCS 2442, pp 61-76, 2002.

[3] G. Ateniese, G. Tsudik, and D. Song. "Quasi-efficient revocation of group signatures," Financial Cryptography 2002.

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

[5] D. Boneh, H. Shacham, "Group Signatures with Verifier-Local Revocation," ACM CCS 2004.

[6] D. Boneh, X. Boyen, H. Shacham, "Short Group Signatures", CRYPTO 2004, LNCS 3152, pp. 41-55.

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

[8] www.iusmentis.com/society/privacy