로밍 유저를 위한 패스워드 기반 키 분배에 관한 연구

이덕규, 이임영

순천향대학교 정보기술공학부

A Study on the Password-based Key distribution for the Roaming User

요 약

네트워크에 대한 접근성이 높아짐으로 인해 사용자는 자신의 클라이언트 터미널을 벗어나 다른 터미널에서의 네트워크 접속을 요구하는 사용자(Roaming User)가 증가하고 있다. 이러한 사용자는 패스워드를 기반으로 하여 현재 위치하고 있는 서버와 새로운 비밀키를 생성할 수 있다. 본 논문에서는 기존의 패스워드 기반의 프로토콜을 간략하게 살펴보고 이를 통하여 프로토콜의 구성 방법과 사전공격 등 여러 가지 일반적인 공격으로부터 안전한 프로토콜을 제안한다. 또한 본 제안 방식은 기존 패스워드를 이용하여 로밍 방식에 적용할 수 있도록 하였다.

I. 서론

인터넷의 활용 영역이 다양한 분야로 확대되면서 어려운 키를 기반으로 하는 방법보다 사용자에게 친숙하게 느낄 수 있는 것으로 서비스를 제공하기 위한 패스워드 기반 통신에 대한 요구가 증가하고 있다. [10]

따라서 안전한 패스워드 기반 통신을 위해 패스워드에 대한 기밀성(Confidentiality)과 무결성(integrity) 그리고 패스워드에 따른 고려사항을 만족해야 한다.

패스워드 기반의 키 교환 프로토콜은 효율적이며 안전한 패스워드 관리기능을 제공해야 한다. 패스워드를 안전하게 보관하기 위하여 각 세션키를 생성할 때 forward secrecy와 세션키에 대한 노출로 인해 패스워드를 보호할 수 있는 backward secrecy를 제공해야 한다.

그리고 패스워드의 효율성(efficiency)과 확장성(scalability)을 고려할 때, 패스워드에 따른 키의 갱신과 효율성에 따른 키의 생성이 쉬워야 한다.

최초로 LGSN[7]에 의해 제안되었고, EKE(Encrypted Key Exchange)에 의해 인증서를 사용하지 않는 DH-EKE와 A-EKE의 2가지 모델을 제안하였다.

GLNS[8]은 LGSN로부터 발전된 프로토콜로서 이를 바탕으로 변형 및 개선된 프로토콜이 많이 제안되었다.

최근 네트워크에 대한 접근성이 높아짐으로 인해 사용자는 자신의 클라이언트 터미널을 벗어나 다른 터미널에서의 네트워크 접속을 요구하는 사용자(Roaming User)가 증가하고 있다. 이러한 사용자는 다른 클라이언트 터미널에서 네트워크에 접근하는 사용자를 의미한다. 여기서 사용자는 패스워드를 기반으로 하여 현재 위치하고 있는 서버와 새로운 비밀키를 생성할 수 있다.

본 논문의 제안방식은 패스워드를 기반으로 최초 서버와 키 교환이 이뤄지고 난 뒤 다른 서버에 접근 하였을 경우 다른 서버는 최초 서버로부터 사용자 정보를 제공 받고 이를 이용하여 사용자 인증 및 키 분배가 이뤄지도록 제안한다.

II. 기존 방식 분석 및 고려 사항

2.1 패스워드의 고려 사항

안전한 통신의 핵심은 패스워드의 관리이며, 패스워드를 관리함에 있어 고려해야 하는 사항으로 패스워드에 대한 안전성과 관리에 다른 효율성을 고려해야 한다.[2][3][6]

· 패스워드의 비밀성

가장 기본적인 성질로서, 어떤 악의적인 공격자가 패스워드를 도출해 내는 것이 계산상 불가능하여야 한다.

· 사전공격(Dictionary attack)

다른 사람으로 가장한 사용자의 반복적인 online상에서의 추측의 위협이 나타날 수 있다. 하나의 패스워드에 대한 불법적인 접근 시도에 대해 접근을 막는 것이 중요하다.

· Forward Secrecy

악의적인 공격자가 이전 세션키에 대한 정보를 알고 있더라도 이후의 세션키를 계산하지 못하게 함으로써 데이터에 접근할 수 없어야 한다.

· Backward Secrecy

악의적인 공격자가 이후에 알려진 세션키에 대한 정보를 가지고서 이전 세션키를 계산하지 못함으로써 데이터에 접근할 수 없어야 한다.

위 네 가지 성질들을 서로 연관성을 가지고 고려되어야 한다. 패스워드를 통해 세션키의 forward secrecy를 제공해야 하며 세션키에 대한 노출로 인해 패스워드를 보호할 수 있는 backward d secrecy를 제공해야 한다. 그리고 패스워드가 노출되는 것을 막기 위해 패스워드의 독립성을 제공해야 한다.

2.2 기존방식에 대한 분석

1) AMP 방식

AMP((Authentication and Key Agreement via Memorable Password))의 전체적인 특징은 확장한 패스워드를 기반으로 한 패스워드 인증을 하고 있다. 이 방식은 또한 인증의 과정에서 키 합의를 이룰 수가 있다 확장한 패스워드 증명은 영지식 증명(Zero-Knowledge Proof)을 이룰 수가 있다.[9]

그림 1은 AMP 전체적인 프로토콜에 대해 설명하고 있다. 이 방식에서 보면 총 5개의 제안방식을 포함하고 있는데 그중 2개의 프로토콜은 패스워드는 서버 위협(server compromise)과 사전공격(Dictionary attack)에 안전하도록 설계되어 있다. 하지만 2개의 프로토콜을 제외한 나머지 3개의 제안 프로토콜이 가지고 있는 문제점으로는 사전공격, server impersonation, client impersonation을 제공하지 못하는 취약점을 가지고 있는데 각각에 대하여 살펴보면 다음과 같다.

Alice (id, π)		Bob (id, g^{π}, π)
$x \in_R Z_q$ $g_1 \in g^x$	$\xrightarrow{(id, g^x)}$	
		$fetch(id, \pi)$ $y \in_R Z_q$
	$\xleftarrow{g^{(x+\pi)y}}$	
$\omega = (x+\pi)^{-1} \bmod q$ $\alpha = (g_2)^{\omega}$ $K_1 = h_1(\alpha)$		$g_2 = (g_1 \cdot g^{\pi})^y$ or $(g_1)^y g^{\pi y}$ $\beta = (g_1)^y$ $K_2 = h_1(\beta)$
$H_{11} = h_2(g_1, K_1)$	$\xrightarrow{h(g_1, K_1)}$	$H_{12} = h_2(g_1, K_1)$ $verify\ H_{11} \doteq H_{12}$
$H_{21} = h_3(g_2, K_1)$ $verify\ H_{21} \doteq H_{22}$	$\xleftarrow{h(g_2, K_2)}$	$H_{22} = h_3(g_2, K_2)$

그림 1: AMP Protocol

사전공격의 경우 패스워드가 사용자에게 위치하고 서버에게는 변형된 π를 제공하기 때문에 사용자가 가지고 있는 패스워드에 대한 사전 공격으로써 사용자 위장공격이 가능하게 된다. 마지막으로 임의의 π를 통한 서버 위장공격도 가능하게 된다. 하지만 패스워드에 대한 추측공격에 대해서는 g^{π}를 안전하게 보관함으로써 개선할 수 있다.

2) A-EKE 방식

본 A-EKE((Authentication and Encrypt ed Key Exchange)방식에서의 특징은 다음과 같다. 사용자는 패스워드 $pwa = password$를 가지며 서버는 사용자에게서 $pwb = f(pwa)$를 가지게 된다. 이러한 패스워드에 대한 분배를 바탕으로 여러 공격에 대해 안전성을 제공한다.[1][4] 공격자가 패스워드를 알고 있다하더라도 이미 분배된 세션키에 대한 정보를 알 수 없다.(forward secrecy 제공) 도청에 대해 안전하도록 설계되어 있다. 쉽게 가할 수 있는 공격에 대해 안전성을 제공하고 있다. 하지만 사용자는 패스워드를 가지고 서버에게는 변형된 패스워드를 제공함으로써 pwb에 대한 취약점이 나타날 수 있다.

III. 제안방식

본 제안 방식에서는 다음과 같이 두 부분으로 나뉘게 된다. 우선 최초 사용자의 정보를 공유하게 될 서버가 존재하게 되고 이후에 사용자의 정보를 제공받게 될 서버로 구성된다. 이때 사용자는 최초 서버와도 키 교환 및 인증이 가능하고 다른 서버와도 키 교환 및 인증이 가능토록 구성하였다. (그림 2 참조)

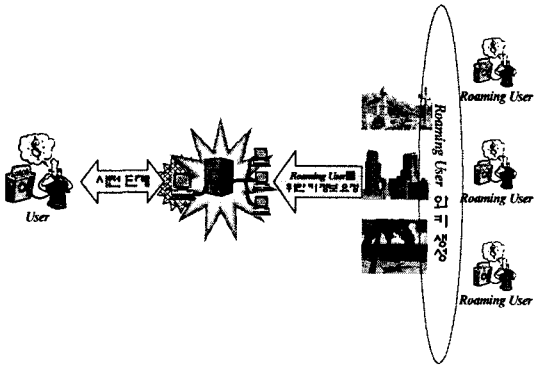

그림 2: 제안 방식 전체 구성도

3.1 시스템 계수

다음은 본 제안방식에서 사용되는 시스템 계수에 대한 설명이다.

A : Client, B : First Server

C : Other Server ID_* : $*$의 식별자

pw : 패스워드

q : 유한체 GF(q)를 정의하는 큰 소수

r : r|q-1인 소수

g : GF(p)상에서 원시원소, m : 공개정보

pb : $pb \equiv g^{pw} \bmod q$로 사용자 A가 생성

E : A와 B에 사용되는 암호화 함수들

E_w^{-1} : A와 B에 사용되는 복호화 함수들

G_A, G_{A+1}, G_B : 사용자 A와 사용자 B가 생성하는 랜덤 값

w : 초기 사용되는 암호화 값

L : 사용자 A가 서버 B에게 검증을 위해 제공하는 해쉬값

Z : 사용자A가 서버 B에게 암호화하여 제공하는 값

Z' : 서버 B가 복호화 하는 값

M : 사용자 A가 서버 C에게 암호화하여 제공하는 값

M' : 서버 C가 복호화 하는 값

$H(\ )$: One-way Hash Function

K : 생성된 Session Key

K' : 생성된 Session Key 검증 값

$\doteq$: 두 값의 비교

3.2 제안 방식

본 제안 방식은 사용자가 서버에게 접속요청 후 서버가 정보를 생성하여 사용자에게 제공하여 키 동의 절차를 받는 과정이 본 논문에서는 서로간의 서버에 대해 신뢰된 채널을 형성하고 있는 것으로 가정한다.

1) 사전 단계

사용자와 서버간의 사전 단계로서 사용자는 패스워드를 서버와 서로 공유하게 되는데 이때 사용자는 pw를 서버는 pb를 공유하게 된다. 또한 $seed$값을 서로 공유함으로써 패스워드에 대한 안전성을 높게 하였다.

2) 최초 서버와 키 교환 과 인증 단계

다음은 키 교환 단계로서 사용자와 서버사이에 세션키를 암호화할 w를 생성하고 이를 바탕으로 암호화하여 전송한다.

step 1. 사용자는 식 (1), 식 (2)와 식 (3)를 생성하고 식(1), (2), (3)을 바탕으로 암호화하여 식 (4)를 생성하고 식 (5)를 전송한다.

$$G_A = g^x \quad (1)$$

$$w = H(pb||seed) \quad (2)$$

$$Z = G_A \cdot H(g^{pw})^{-1} \quad (3)$$

$$L = E_w(Z) \quad (4)$$

$$(ID_A, L) \quad (5)$$

step 2. 서버는 식 (5)를 받아 이를 복호화 하여 Z를 획득한다. 획득한 Z와 자신이 가지고 있는 pb를 이용하여 식 (8)을 이용하여 서버가 생산한 식 (7)과 식 (8)로서 세션키(식 (9))를 생성한다. 생성된 세션키 K, γ, Z를 해쉬하여 사용자에게 전송한다.(식 (12))

$$L' = E_w^{-1}(E_w^1(Z)) = Z \quad (6)$$

$$G_B = g^y \quad (7)$$

$$\gamma = pb \cdot Z$$
$$= g^{pw} \cdot H(g^{pw}) \cdot g^x \cdot H(g^{pw})^{-1}$$
$$= g^{pw+x} \quad (8)$$

$$K = (\gamma)^y \quad (9)$$

$$\alpha = H(Z, \gamma, K) \quad (10)$$

$$P = E^2(G_B) \quad (11)$$

$$(ID_B, P, \alpha) \quad (12)$$

step 3. 사용자는 미리 γ'를 계산(식 (13))하고 서버로부터 받은 P로부터 G_B를 계산(식 (14))하여 세션키(식 (15))를 생성하고 α를 검증(식 (16), (17))한다. 사용자는 식 (18)을 생성하고 식 (19)를 해쉬하여 서버에게 전송(식 (19))한다.

$$\gamma' = g^{pw} \cdot g^x = g^{pw+x} \quad (13)$$

$$P' = E_w^{-2}(E_w^2(G_B)) = G_B \quad (14)$$

$$K' = (G_B)^{pw+x} \quad (15)$$

$$\alpha' = H(Z, \gamma, K) \quad (16)$$

$$verify\ \alpha \doteq \alpha' \quad (17)$$

$$\delta = (G_B)^w \quad (18)$$

$$\beta = H(G_B, \delta, K) \quad (19)$$

step 4. 서버는 미리 δ값을 생성(식 (20))하고 사용자로부터 받은 β값을 검증한다.

$$\delta' = (G_B)^w \quad (20)$$

$$\beta' = H(G_B, \delta, K) \quad (21)$$

$$verify\ \beta \doteq \beta' \quad (22)$$

3) 다른 서버와 키 교환 및 인증 단계

step 1. 사용자는 다른 서버에 접근하여 (식 (23))을 생성하고 다음(식 (24))을 암호화하여 전송(식 (25))를 전송한다

$$M = G_{A0} \cdot H(w)^{-1} \quad (23)$$

$$W = H(w)$$

$$X = E_W(M) \quad (24)$$

$$(ID_A \mid ID_C \mid X) \quad (25)$$

step 2. 다른 서버는 전송 받은 값 중에서 ID 값을 이용하여 사용자의 w을 이용하여 다음을 복호화 한다.

$$B \rightarrow C : w \quad (26)$$

$W0 = H(w)$

$$M' = E_W^{-1}(E_W^1(M)) = M \quad (27)$$

$$G_C = g^y \quad (28)$$

$$\gamma' = w \cdot M \quad = H(W) \cdot g^x \cdot H(W)^{-1}$$

$$= g^x \quad (29)$$

$$K' = (\gamma')^y \quad (30)$$

$$\alpha' = H(M, \gamma', K') \quad (31)$$

$$P' = E_W^2(G_C) \quad (32)$$

$$(ID_C, P', \alpha') \quad (33)$$

step 3. 다른 서버는 사용자 인증과정을 완료하고 사용자와의 키를 생성하고 다시 전송한다

$$\gamma'' = g^x \quad (34)$$

$$P'' = E_W^{-2}(E_W^2(G_C)) = G_C \quad (35)$$

$$K'' = (G_C)^x \quad (36)$$

$$\alpha'' = H(M, \gamma', K') \quad (37)$$

$$verify\ \alpha' \doteq \alpha'' \quad (38)$$

$$\delta' = (G_C)^W \quad (39)$$

$$\beta' = H(G_C, \delta', K'') \quad (40)$$

step 4. 사용자는 최종적으로 다른 서버를 인증하고 키를 생성하고 사용한다.

$$\delta'' = (G_C)^W \quad (41)$$

$$\beta'' = H(G_C, \delta'', K') \quad (42)$$

$$verify\ \beta' \doteq \beta'' \quad (43)$$

IV. 제안 방식 분석

본 방식에서 제안한 패스워드 기반 키 분배 프로토콜에 대해 살펴보면 다음의 특징을 볼 수 있다.

서버 위장 공격(Server Impersonation) : 클라이언트와 서버 사이에 안전하게 $pb \equiv g^{pw} \bmod q$ 를 서로 공유하게 되므로, 클라이언트가 요청하고 서버가 보내는 경우와 사용자가 $pb \equiv g^{pw} \bmod q$를 이용하여 서버에게 메시지를 전송할 때, 공격자는 $pb \equiv g^{pw} \bmod q$를 모르고 있기 때문에 서버 위장 공격에서 안전할 수 있다. 또한 사용자와 함께 초기 서버는 *seed*값을 공유하게 됨으로써 서버와 사용자 이외에 패스워드가 불법적으로 노출된다하더라도 w대해 안전할 수 있다.

클라이언트 위장 공격(Client Impersonation) : 클라이언트 위장 공격은 서버 위장 공격과 마찬가지로 $pb \equiv g^{pw} \bmod q$에 대한 것은 공격자가 모르기 때문에 클라이언트 위장 공격에 안전하다. 만약 공격자가 임의의 pb를 생성하여 전송하더라도 원 pw에 대한 pb가 아니므로 클라이언트는 취소할 수 있다.

사전 공격(Dictionary Attack) : pw에 대한 사전 공격은 클라이언트가 임의적인 pw를 생성한 것이 아니라, 사전의 조합으로 생성되었다면 어느 정도 신뢰성을 잃을 수 있다. 하지만 본 제안 방식에서는 pw를 이용하는 것이 아니라 pw를 바탕으로 하여 pb를 생성하고 이것을 이용하여 통신하기 때문에 안전하다고 할 수 있다. 그에 따른 증명은 hard-problem에 기초하고 있다.

추측공격에는 두 가지 관점에서 볼 수 있는데 첫째는 On-line Guessing Attack으로 pw에 대한 온라인 추측 혹은 session key K에 대한 추측 공격으로 나뉘어 볼 수 있다. pw 온라인 추측 공격은 통신이 이뤄지는 당시에 pb에 대한 추측 공격으로 pb는 hard-problem에 기초하고 있어 온라인 공격에서는 안전하다. session key K에 대한 추측 공격은 다음에서 언급할 PFS와 Backward Secrecy에서 다루도록 한다.

둘째는 Off-line Guessing Attack으로써 오프라인 추측 공격도 pw와 K에 대하여 나뉘어 볼 수 있는데 pw에 대한 공격은 사전공격과 동일시 할 수 있으며, K에 대하여서는 기존 session key K가 노출된다 하더라도 이후 key 생성에 아무 것도 알 수 없다.

Perfect Forward Secrecy : 이전 세션키를 알더라도 다음 세션키를 생성할 수 없어야 한다. 이전

표 1: 기존 방식과 제안 방식의 비교 분석

	사전 공격	패스워드 추측공격	Forward secrecy	backward secrecy	server compromise	Roaming Service
AMP	×	○	○	○	×	×
A-EKE	○	×	○	○	×	×
제안 방식	○	○	○	○	○	○

세션키 생성과 이후 세션키 설정에 연관성이 없기 때문에 이전 세션키를 알더라도 이후 세션키를 알 수 없다. 또한 이전 세션키가 다음 세션키에 영향을 주지 않는다. $G_A, G_{A'}, G_B$가 매 세션마다 새로이 생성됨으로 이전 $G_A, G_{A'}, G_B$를 알았다 하더라도 이후 세션키에는 영향을 못 미친다.

제안 방식과 기존의 방식과의 살펴보면 표 1과 같은 결과를 얻을 수 있다.

V. 결론

어느 곳에서 자유롭게 네트워크에 쉽게 접근 할 수 있게 됨에 따라 사용자는 자신의 클라이언트 단말기를 벗어나 다른 단말기를 이용하여 네트워크에 접속하는 일이 자주 발생하고 있다. 하지만 단말기의 공유에 따라 사용자의 정보가 유출되는 등의 심각한 문제를 발생한다. 이러한 이유로 사용자의 비밀정보는 단말기에 저장되지 않고 안전하게 보관되고 접근할 수 있는 방법이 필요하게 된다.

패스워드 기반의 키 분배 프로토콜은 위에서 언급한 필요성을 만족시킬 수 있는 방법으로 제안하였다. 본 제안방식은 사용자에 대한 인증을 통한 뒤 새롭게 서버와 키를 생성하는 방식을 택함으로써 다른 서버에는 자신의 정보를 숨기게 되며 최초 서버 또한 사용자의 실제 패스워드 정보를 알고 있지 않기 때문에 최초 서버가 다른 서버와 같이 공모와 같은 불법 행위에 대해 안전하도록 설계하였다.

본 논문에서 제공하고 있는 프로토콜은 패스워드에 대한 안전성을 서버에 다른 값을 전달함으로써 실현하였고, 초기 사전공격이나 추측공격을 예방하기 위해 pb값을 이용하여 공격에 대한 안전성을 부여하였다.

키 교환 프로토콜은 인증과 키 교환 분야는 많은 분야에서 사용되고 있는 실정이다. 이를 토대로 자신의 패스워드를 분실하였을 경우 이를 해결할 수 있는 키 복구와 같은 분야에서 많은 연구가 필요하리가 생각한다.

참고문헌

[1] M. Ballare, D. Pointcheval, and P. Rogaway. Authenticated key exchange secure against dictionary attack, In EUROCRYPT 2000

[2] S. Bellovin and M. Merritt. Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks. Proc. of Symposium on Security and privacy. page 72-84. IEEE, 1992

[3] S. Bellovin and M. Merritt. Augmented Encrypted Key Exchange : A Password-Based Protocol Secure against dictionary Attacks and Password File Compromise. Proceeding of the 1st Annual Conference on Computer and Communications Security, ACM, 1993

[4] V. Boyko, P. Mackenzie and S. Patel. Provably Secure Password Authenticated Key Exchange Using Diffie-Helllman. To appear in Eurocypt 2000

[5] D. Jablon. Strong Password-Only Authenticated Key Exchange. ACM Computer Communications Review, October 1996

[6] P. Mackenzie and R. Swaminathan, Secure network authentication with Password identification, Presented to IEEE p1363a, August 1999

[7] S. Lucks, Open Key Exchange: How to defeat Dictionary Attacks without encrypting public-keys, The Security Protocol Workshop '97, April 7-9, 1997

[8] M. Lomas, L. Gong, J. Saltzer, and R, Needham, Reducing risks from poorly chosen keys, Proceedings of the 12th ACM Symposium on Operating System principles, ACM Operating Systems Review, 1989, pp. 14-18

[9] L. Gong, M. Lomas, R. Needham, and J. Saltzer, Protecting poorly chosen secrets from guessing attacks, IEEE journal on SAC., vol. 11, no.5, pp.648-656, June 1993

[10] T. kwon, Ultimate solution to authentication via memorable password