제19회 한국정보처리학회 추계학술발표대회 논문집 제9권 제2호 (2003.05)

안전하고 효율적인 A-EKE에 관한 연구

이덕규*, 이임영*
*순천향대학교 정보기술공학부
e-mail : hbrhdbr@catholic.or.kr, imylee@sch.ac.kr

A Study on Authentication and Encrypted Key Exchange with Security and Efficiency

Deok-Gyu Lee*, Im-Yeong Lee*
*Division of Information Technology, Soonchunhyang University

요 약

고전적인 암호 프로토콜은 사용자-선택키를 기반으로 하였다. 하지만 이러한 방법은 공격자에게 패스워드-예측 공격을 허용하는 문제점을 가지고 있다. 기존에 제안된 방식들은 패스워드에 대한 보호를 강화함으로써 패스워드를 보호하여 하였다. 이러한 문제점으로부터 안전하지 못한 네트워크 상에서 사용자를 인증하고 서로간의 세션키를 공유하는 새로운 방법을 제안한다. 제안된 프로토콜은 능동적인 공격자에 의한 사전공격(Dictionary attack), 패스워드 추측 공격, forward secrecy, server compromise, client compromise와 세션키 분실에 안전하게 설계되었다.

1. 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.
고전 암호 프로토콜에서는 사용자가 키를 선택하는 방식을 기반으로 하였다.(User-Chosen Key) 그러나 이러한 방법은 공격자에게 패스워드-예측 공격을 허용하는 문제점을 가지고 있다.
최초로 LGSN[7]에 의해 제안되었고, EKE(Encrypt ed 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.
제안하는 방식은 안전하고 효율적으로 제공하기 위해 다음을 포함하여 안전하게 설계한다
- 패스워드 기반의 인증
- Diffie-Hellman 기반의 키 동의
- 쉬운 세션키 생성
- 패스워드 파일에 대한 보호

2. 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 secrecy를 제공해야 한다. 그리고 패스워드가 노출되는 것을 막기 위해 패스워드의 독립성을 제공해야 한다.

2.2 기존방식에 대한 분석

1) AMP(Authentication and Key Agreement via

Memorable Password) 방식
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.
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^x를 안전하게 보관함으로써 개선할 수 있다.

2) A-EKE (Authentication and Encrypted Key Exchange) 방식
본 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.

3. 제안방식

본 제안 방식에서는 크게 두 가지의 제안 방식을 설명한다. 첫 번째는 사용자가 서버에게 정보를 제공하는 경우와 사용자의 접속요청 후 서버가 사용자에게 정보를 제공하는 경우이다.

3.1 시스템 계수

다음은 본 제안방식에서 사용되는 시스템 계수에 대한 설명이다.
A : Client, B : 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
$H(\ )$: One-way Hash Function
K : 생성된 Session Key
K' : 생성된 Session Key 검증 값
$\doteq$: 두 값의 비교

3.2 제안 방식(I)

본 제안 방식은 사용자가 서버에게 접속요청 후 서버가 정보를 생성하여 사용자에게 제공하여 키 동의 절차를 받는 과정이다. (그림 2)

1) 사전 단계
사용자와 서버간의 사전 단계로서 사용자는 패스워드를 서버와 서로 공유하게 되는데 이때 사용자는 pw를 서버는 pb를 공유하게 된다.

2) 키 교환 과 인증 단계
다음은 키 교환 단계로서 사용자와 서버사이에 세션키를 암호화할 w를 생성하고 이를 바탕으로 암호화하여 전송한다.

step 1. 서버는 식 (1-1), (1-2), (1-3), (1-4)을 생성하고 식 (1-3)을 바탕으로 암호화하여 식 (1-6)을 생성하고 식 (1-7)을 전송한다.

$$G_A = pb \equiv g^{pw} \bmod q \qquad (1\text{-}1)$$

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

그림 1. AMP Protocol

$G_B \equiv g^y \bmod q$ (1-2)
$w \equiv H((pb)^m \bmod q) \equiv H(g^{pw+m} \bmod q)$(1-3)
$K \equiv G_A \cdot G_B \bmod q \equiv g^{pw+y} \bmod q$ (1-4)
$L = H(\omega \parallel K)$ (1-5)
$Z = E_\omega^1(G_B \parallel L)$ (1-6)
(ID_A, m, Z) (1-7)

step 2. 사용자는 식 (1-7)을 받아 이를 복호화하여 $G_B' \parallel L'$을 획득한다. 획득한 $G_B' \parallel L'$과 자신이 가지고 있는 pw를 이용하여 식 (1-8)을 생성하고, 식 (1-10), (1-11)을 생성하고, 식 (1-12)을 검증한다.
$w' \equiv H((pb)^m \bmod q) \equiv H(g^{pw+m} \bmod q)$(1-8)
$Z' = E_w^{-1}(E_w^1(G_B \parallel L)) = G_B' \parallel L'$ (1-9)
$K' \equiv G_A \cdot G_B' \bmod q \equiv g^{pw+y} \bmod q$ (1-10)
$L' = H(K' \parallel \omega')$ (1-11)
$verify\ L \doteq L'$ (1-12)

3.3 제안 방식(II)

본 제안 방식은 접속시에 사용자가 정보를 생성하여 서버에게 제공하여 키 동의 절차를 받는 과정이다. (그림 3)

1) 사전 단계
사용자와 서버간의 사전 단계로서 사용자는 패스워드를 서버와 서로 공유하게 되는데 이때 사용자는 pw를 서버는 pb를 공유하게 된다.

2) 키 교환 과 인증 단계
다음은 키 교환 단계로서 사용자와 서버사이에 세션키를 암호화할 w를 생성하고 이를 바탕으로 암호화하여 전송한다.

step 1. 사용자는 식 (2-1), (2-2), (2-3), (2-4)을 생성하고 식 (2-3)을 바탕으로 암호화하여 식 (2-6)을 생성하고 식 (2-7)을 전송한다.
$G_A = pb \equiv g^{pw} \bmod q$ (2-1)
$G_{A+1} \equiv g^x \bmod q$ (2-2)
$w \equiv H((pb)^m \bmod q) \equiv H(g^{pw+m} \bmod q)$(2-3)
$K \equiv G_A \cdot G_{A+1} \bmod q \equiv g^{pw+x} \bmod q$ (2-4)
$L = H(\omega \parallel K)$ (2-5)
$Z = E_\omega^1(G_{A+1} \parallel L)$ (2-6)
(ID_A, m, Z) (2-7)
step 2. 사용자는 식 (2-7)을 받아 이를 복호화하여 $G_B' \parallel L'$을 획득한다. 획득한 $G_B' \parallel L'$과 자신이 가

A (pw)	공개 정보 (q, g, r)	B (pb)
	request	$G_A = pb \equiv g^{pw} \bmod q$ $G_B \equiv g^y \bmod q$ $w \equiv H((pb)^m \bmod q) \equiv H(g^{pw+m} \bmod q)$ $K \equiv G_A \cdot G_B \bmod q \equiv g^{pw+y} \bmod q$ $L = H(\omega \parallel K)$ $Z = E_\omega^1(G_B \parallel L)$
$w' \equiv H((pb)^m \bmod q) \equiv H(g^{pw+m} \bmod q)$ $Z' = E_w^{-1}(E_w^1(G_B \parallel L)) = G_B' \parallel L'$ $K' \equiv G_A \cdot G_B' \bmod q \equiv g^{pw+y} \bmod q$ $L' = H(K' \parallel \omega')$ $verify\ L \doteq L'$	(ID_A, m, Z)	

그림 2. 제안방식 I

A (pw)	공개 정보 (q, g, r)	B (pb)
$G_A = pb \equiv g^{pw} \bmod q$ $G_{A+1} \equiv g^x \bmod q$ $w \equiv H((pb)^m \bmod q) \equiv H(g^{pw+m} \bmod q)$ $K \equiv G_A \cdot G_{A+1} \bmod q \equiv g^{pw+x} \bmod q$ $L = H(\omega \parallel K)$ $Z = E_\omega^1(G_{A+1} \parallel L)$	(ID_A, m, Z)	$w' \equiv H((pb)^m \bmod q) \equiv H(g^{pw+m} \bmod q)$ $Z' = E_w^{-1}(E_w^1(G_B \parallel L)) = G_B' \parallel L'$ $K' \equiv G_A' \cdot G_{A+1}' \bmod q \equiv g^{pw+x} \bmod q$ $L' = H(K' \parallel \omega')$ $verify\ L \doteq L'$

그림 3. 제안 방식 II

지고 있는 pb를 이용하여 식 (2-10)을 생성하고, 식 (2-10), (2-11)을 생성하고, 식 (2-12)을 검증한다..

$$w' \equiv H((pb)^m \bmod q) \equiv H(g^{pw+m} \bmod q) \quad (2\text{-}8)$$

$$Z = E_w^{-1}(E_w^1(G_B \parallel L)) = G_B' \parallel L' \quad (2\text{-}9)$$

$$K' \equiv G_A' \cdot G_{A+1}' \bmod q \equiv g^{pw+x} \bmod q \quad (2\text{-}10)$$

$$L' = H(K' \parallel \omega') \quad (2\text{-}11)$$

$$verify\ L \doteq L' \quad (2\text{-}12)$$

4. 제안 방식 분석

본 방식에서 제안한 A-EKE 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.

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 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+1}, G_B가 매 세션마다 새로이 생성됨으로 이전 G_A, G_{A+1}, G_B를 알았다 하더라도 이후 세션키에는 영향을 못 미친다.

전체적으로 1-flow이며, 제안방식에서의 가장 큰 메시지는 Z가 된다. 나머지는 해쉬 함수를 통한 값을 전송하게 됨으로 적은 메시지를 통해 인증의 문제를 해결하였다. 또한 사용자와 서버간의 암호화는 2번으로 진행되며 전체적으로 exponential 연산의 경우 각각 3번의 연산으로 이뤄지도록 하였다. 여기서 g^x와 g^{pw}를 미리 계산하게 되면 사용자는 2번, 서버는 3번의 연산으로 줄일 수 있다. 제안 방식과 기존의 방식과의 살펴보면 표 1과 같은 결과를 얻을 수 있다.

표 1. 기존 방식과 제안 방식의 비교 분석

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

5. 결론

본 논문에서는 패스워드를 기반한 안전한 인증 및 키 교환을 위한 프로토콜을 제안하였다. 사용자들의 패스워드를 기반한 프로토콜이 갖는 사전공격, 추측공격 등에 효율적인 프로토콜을 설계하였다.

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

키 교환 프로토콜은 인증과 키 교환 분야는 많은 분야에서 사용되고 있는 실정이며, 키 로밍(Key roaming) 많은 분야에서 이용될 것이다. 이러한 관계에 따른 키-교환 프로토콜에 많은 연구가 진행할 계획이다.

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