

패스워드를 이용한 키 분배에 관한 연구

이덕규, 이임영
순천향대학교 정보기술공학부

A Study on Key exchange using password

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

요 약

고전적인 암호 프로토콜은 사용자-선택 키를 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(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 제안되었

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-pass 구현

2. 기존 방식 분석 및 고려 사항

1 패스워드의 고려 사항

안전한 통신의 핵심은 패스워드의 관리이며, 패스워드를 관리함에 있어 고려해야 하는 사항으로 패스워드에 대한 안전성과 관리에 다른 효율 성을 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 통한 세션키에 대한 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. 기존방식에 대한 분석

1) AMP(Authentication and Key Agreement via Memorable Password)

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 패스워드는 서버 위협(server compromise)과 사전공격(Dictionary attack)에 안전하도록 설계되어 있다. 하지만 AMP에서 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 π 를 제공하기 때문에 사용자가 가지고 있는 패스워드에 대한 사전 공격으로써 사용자 위장공격이 가능하게 된다. 마지막으로 임의의 π 를 통한 서버 위장공격도 가능하게 된다. 하지만 패스워드에 대한 추측공격에 대해서는 g^x 를 안전하게 보관함으로써 개선할 수 있다.

2) A-EKE (Authentication-Encrypted Key Exchan-

ge)

본 방식에서의 특징은 다음과 같다. 사용자는 패스워드 $pwa = password$ 를 가지며 서버는 사용자에게서 $pub = f(pwa)$ 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 pub 에 대한 취약점이 나타날 수 있다.

3. 제안 방식

다음은 기존의 패스워드를 기반으로 한 AMP와 A-EKE의 문제점에 대한 해결방안으로 본 방식을 제안한다. 제안 방식은 비대칭적 신뢰모델을 가정하게 되는데 Client A는 패스워드 pw 를 가지고 Server B는 pw 의 특별한 값인 pb 를 가진다. 프로토콜의 두 개체들은 Diffie-Hellman 키 교환을 이용하고 있다

1. 시스템 계수

다음은 본 제안방식에서 사용되는 시스템 계수에 대한 설명이다.

- A : Client, B : Server
- ID_i : *의 식별자, pw : 패스워드
- q : 유한체 GF(q)를 정의하는 큰 소수
- r : $rlq-1$ 인 소수, g : GF(p)상에서 원시원소
- m : 공개정보
- pb : $pb \equiv g^{pw} \pmod q$ 로 사용자 A가 생성
- E^1, E^2 : A와 B에 사용되는 암호화 함수들
- E_w^{-1}, E_w^{-2} : A와 B에 사용되는 복호화 함수들
- G_A, G_B : 사용자 A와 사용자 B가 생성하는 랜덤 값

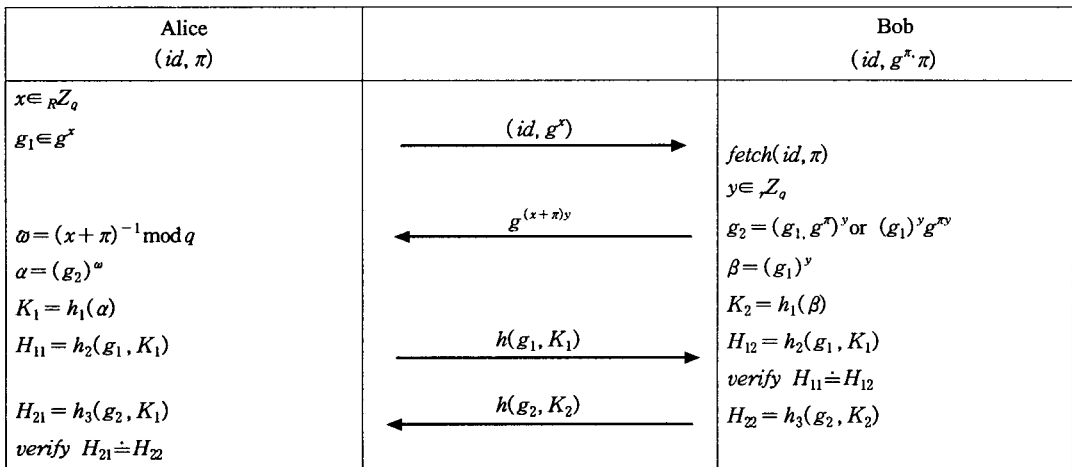


그림 1. AMP Protocol

w : 초기 사용되는 암호화 값
 ρ_A, δ : 사용자 A가 서버 B에게 제공하는 값
 α, α' : 키 생성을 위한 값
 L : 사용자 A가 서버 B에게 암호화하여 제공하는 값
 L' : 서버 B가 복호화하는 값
 $H(\)$: One-way Hash Function
 ρ_B : 서버 B가 생성하여 사용자 A에게 제공하는 값
 K : 생성된 Session Key
 K' : 생성된 Session Key 검증 값
 \mathcal{E} : 서버 B가 사용자 A에게 제공되는 값
 \doteq : 두 값의 비교

식 (2)를 바탕으로 암호화하여 식 (6)를 생성하고 식 (7)를 전송한다.

$$\begin{aligned}
 G_A &\equiv g^x \pmod q & (1) \\
 w &\equiv H(g^{pw+m} \pmod q) & (2) \\
 \rho_A &\equiv g^{x-pw} \pmod q & (3) \\
 \alpha &\equiv (\rho_A)^m \pmod q & (4) \\
 \delta &= H(\alpha \parallel \rho_A) & (5) \\
 L &= E_w^1(\rho_A \parallel \delta) & (6) \\
 &(ID_A, L) & (7)
 \end{aligned}$$

2. 제안 프로토콜

다음은 각 단계에 대하여 자세히 기술하도록 하겠다.

1) 사전 단계

사용자와 서버간의 사전 단계로서 사용자는 패스워드를 서버와 서로 공유하게 되는데 이때 사용자는 pw 를 서버는 pb 를 공유하게 된다. 또한 공개된 m 값을 이용하여 패스워드에 대한 안전성을 높게 하였다.

2) 키 교환 과 인증 단계

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

step 2. 서버는 식 (7)를 받아 이를 복호화하여 $\rho_A' \parallel \delta'$ 를 획득한다. 획득한 $\rho_A' \parallel \delta'$ 와 자신이 가지고 있는 pb 를 이용하여 식 (11) (13)을 검증하고 식 (11)을 이용하여 서버가 생산한 식 (15)로서 세션키(식 (16))를 생성한다. 생성된 세션키 K, ρ_B 를 해쉬하고 암호화하여 사용자에게 전송한다.(식 (19))

$$\begin{aligned}
 G_B &\equiv g^y \pmod q & (8) \\
 w &\equiv H((pb)^m \pmod q) \equiv H(g^{pw+m} \pmod q) & (9) \\
 L' &= E_w^{-1}(E_w^1(\rho_A \parallel \delta)) = \rho_A' \parallel \delta' & (10) \\
 \alpha' &\equiv (\rho_A)^m \pmod q & (11) \\
 \delta' &= H(\alpha' \parallel \rho_A') & (12) \\
 \text{verify } \delta &\doteq \delta' & (13) \\
 \rho_B &\equiv pb \cdot G_B \pmod q \equiv g^{pw+y} \pmod q & (14) \\
 K &\equiv \alpha' \cdot \rho_B \pmod q \equiv g^{x-pw} \cdot g^{y+pw} \pmod q
 \end{aligned}$$

step 1. 사용자는 식 (1), (2), (3), (4)를 생성하고 식

A (pw)	공개 정보 (q, g, r)	B (pb)
$G_A \equiv g^x \pmod q$ $w \equiv H(g^{pw+m} \pmod q)$ $\rho_A \equiv g^{x-pw} \pmod q$ $\alpha \equiv (\rho_A)^m \pmod q$ $\delta = H(\alpha \parallel \rho_A)$ $L = E_w^1(\rho_A \parallel \delta)$	(ID_A, L)	$G_B \equiv g^y \pmod q$ $w \equiv H((pb)^m \pmod q) \equiv H(g^{pw+m} \pmod q)$ $L' = E_w^{-1}(E_w^1(\rho_A \parallel \delta)) = \rho_A' \parallel \delta'$ $\alpha' \equiv (\rho_A)^m \pmod q$ $\delta' = H(\alpha' \parallel \rho_A')$ $\text{verify } \delta \doteq \delta'$ $\rho_B \equiv pb \cdot G_B \pmod q \equiv g^{pw+y} \pmod q$ $K \equiv \alpha' \cdot \rho_B \pmod q \equiv g^{x-pw} \cdot g^{y+pw} \pmod q$ $\equiv g^{(x+y)} \pmod q$ $\tau = H(K \parallel \rho_B)$
$\mathcal{E} = E_w^{-2}(E_w^2(\rho_B \parallel H(K \parallel \rho_B)))$ $= K \parallel \rho_B$ $K' \equiv \rho_B' \cdot \rho_A \pmod q$ $\equiv g^{y+pw} \cdot g^{x-pw} \pmod q \equiv g^{x+y} \pmod q$ $\text{verify } \tau \doteq \tau'$	(ID_B, \mathcal{E})	$\mathcal{E} = E_w^2(\rho_B \parallel \tau)$

그림 2. 제안 방식

$$\equiv g^{(x+y)} \pmod q \quad (15)$$

$$\tau = H(K \parallel \rho_B) \quad (16)$$

$$\mathcal{E} = E_w^2(\rho_B \parallel \tau) \quad (17)$$

$$(ID_B, \mathcal{E}) \quad (18)$$

step 3. 사용자는 서버로부터 받은 \mathcal{E} 로부터 ρ_B 를 이용하여 K 를 계산하여 세션키(식 (21))를 생성하고 τ 를 검증(식 (22))한다.

$$\begin{aligned} \mathcal{E}' &= E_w^{-2}(E_w^2(\rho_B \parallel H(K \parallel \rho_B))) \\ &= K \parallel \rho_B \end{aligned} \quad (19)$$

$$\begin{aligned} K' &\equiv \rho_B' \cdot \rho_A \pmod q \\ &\equiv g^{y+pw} \cdot g^{x-pw} \pmod q \equiv g^{x+y} \pmod q \end{aligned} \quad (20)$$

$$\text{verify } \tau \doteq \tau' \quad (21)$$

4. 제안 방식 분석

본 방식에서 제안한 AEKE 프로토콜에 대해 살펴보면 다음의 특징을 볼 수 있다.

- Perfect forward secrecy : pb 가 노출되어도 공격자는 pw 를 알수 없기 때문에, 세션키를 찾을 수가 없다. 왜냐하면 hard problem을 풀 수 없기 때문이다.
- backward secrecy : 세션키가 노출되어도 공격자는 패스워드 파일을 찾을 수가 없다. 또한 세션키 생성에 사용되어지는 ω , pw 를 찾을 수 없다.
- Replay attack : L, \mathcal{E} 는 사용자와 서버쪽에서 세션키 생성을 위해 사용되는 계수으로써 내부에 생성된 값은 사용자와 서버가 매 세션마다 새로이 생성하기 때문에 공격자가 다시 사용하는 것은 불가능하다.
- off-line guessing attack : 공격자가 L 에 대해 분해할 수가 없으므로 불가능하며, man-in-the-middle attack 역시 α, ρ_B 값을 알 수 없으므로 막을 수 있다.

표 1. 기존 방식과 제안 방식과 비교 분석

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

전체적으로 2-flow이며, 제안방식에서의 가장 큰 메시지는 L 과 \mathcal{E} 가 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 5번의 연산으로 이뤄지도록 하였다. 여기서 g^x 와 g^{pw} 를 미리 계산하게 되면 사용자는 4번, 서버는 4번의 연산으로 줄일 수 있다. 제안 방식과 기존의 방식과의 살펴보면 표 1과 같은 결과를

얻을 수 있다.

5. 결론

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

본 논문에서 제공되는 forward secrecy, backward secrecy, replay attack, guessing attack등 취약성과 안전성을 제공하고 있다.

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) 혹은 키 복구(Key recovery)등 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. Bellare 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. Bellare 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-Hellman. To appear in Eurocrypt 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
- [11] 최용락, 소우영, 이계광, 이임영, 컴퓨터 통신 보안, 도서출판 그린, 2001. 2