일회용 패스워드 기반의 키 교환 프로토콜

서승현^{O} 조태남 이상호
이화여자대학교 컴퓨터학과 컴퓨터
{happyday^{O}, tncho, shlee}@ewha.ac.kr

A New Key Exchange Protocol based on One-Time-Password

SeungHyun Seo^{0} TaeNam Cho SangHo Lee
Dept. of Computer Science & Engineering, Ewha Womans University

요 약

키 교환 프로토콜에서 상호 인증은 필수 요소이며, 사용자에게 편리하고 비용이 적게 드는 패스워드 기반의 인증 방식이 널리 사용되고 있다. 패스워드 기반의 프로토콜은 패스워드가 가지는 제약으로 인한 공격에 대해서 안전해야 할 뿐 아니라, 사용자의 작업량을 줄이기 위한 효율성도 매우 중요한 요건이다. 본 논문에서는 서버와 사용자간의 인증을 제공하고 세션키를 공유하기 위한 키 교환 프로토콜 OTP-EKE(One Time Password based Encrypted Key Exchange)를 제안한다. 키 교환을 위한 사용자 인증은 패스워드 방식을 채택하였으며, 특히 서버 디렉토리에 대한 공격 등에 대해서 안전도를 높이기 위하여 일회용 패스워드 확인자와 서버의 공개 패스워드를 이용하였다. 제안한 프로토콜은 모듈라 지수승 계산 횟수와 메시지 전송 횟수를 줄임으로써 효율성 향상을 보인다.

1. 서론

공개된 인터넷을 통하여 안전한 통신을 하기 위해서는 전송될 정보를 암호화하여야 한다. 이를 위해서 통신 상대자간에는 공통으로 사용할 수 있는 키의 공유와 서로의 신원을 확인하는 인증 과정이 필요하다.

특히, 사용자 인증 방식 중에서 패스워드와 같이 사용자가 알고 있는 지식을 통한 인증 방식은 쉽게 사용될 수 있어 많이 이용된다. 그러나 사용자들은 쉽게 기억할 수 있는 패스워드를 선택하는 경향이 있기 때문에, 패스워드 자체를 암호화하는 키로 사용한다면 패스워드 추측 공격을 당할 위험이 있고 암호 시스템 자체를 취약하게 만들 수도 있다. 이를 보안하고 안전한 키 공유도 동시에 하기 위해서 패스워드 기반 키 교환 프로토콜들이 인증 및 키 교환 프로토콜로 사용되고 있다[12]. 이러한 프로토콜들은 평문등가 프로토콜(Plaintext-equivalent protocol)과 확인자 기반 프로토콜(verifier-based protocol)로 나눌 수 있으며, 서버가 공격당하더라도 확인자 기반 프로토콜은 패스워드 확인자만이 공격자에게 노출되기 때문에 평문등가 프로토콜에 비하여 더 안전하다고 할 수 있다.

본 논문에서 제안하는 패스워드 기반 키 교환 프로토콜 OTP-EKE(One Time Password based Encrypted Key Exchange)는 확인자 기반 방식의 프로토콜로서 일회용 패스워드 S/Key[6] 방식을 적용하여 효율성을 향상시켰다.

본 논문의 구성은 다음과 같다. 2장에서 용어 및 보안 요구사항을 기술한다. 3장에서 제안한 OTP-EKE 프로토콜을 제시한다. 4장에서 OTP-EKE의 안전성 분석과 효율성을 분석을 하고 5장에서 결론을 맺는다.

2. 개요

이 장에서는 용어 정의와 패스워드 기반 키 교환 프로토콜들이 만족해야하는 기본적인 보안 요구사항들을 기술한다[1][7].

■ **용어 정의**

- A : 사용자
- B : 서버
- ID : 사용자 A의 식별자(ID, identifier)
- pwd : 사용자 A의 패스워드(password)
- $H()$: 일방향 해쉬 함수(one-way hash function),
  $H^i(x) = H(H^{i-1}(x))$
- $E_K()$: 키 K를 사용한 대칭키 암호화 알고리즘
- g : 곱셈 군(multiplicative group) Z_p^*의 생성자(generator)
- p, q : 강한 소수(stong prime), $p=2*q+1$

■ **보안 요구사항**

① 도청 공격에 안전해야 한다.
② 재전송 공격(replay attack)에 안전해야 한다.
③ 중간자 공격(man-in-the-middle attack)에 안전해야 한다.
④ 오프라인 패스워드 추측 공격에 안전해야 한다.
⑤ Denning-Sacco 공격에 안전해야 한다.
⑥ PFS(Perfect Forward Secrecy)를 만족해야 한다.

3. OTP-EKE 프로토콜

기존의 EKE 부류의 확인자 기반방식 프로토콜 A-EKE

B-EKE, AuthA는 2장에서 기술한 보안 요구사항들을 만족하지만, 인증 방식에서 각각 다음과 같은 비효율성을 가진다. A-EKE는 사용자의 패스워드로부터 유도된 공개키와 비밀키를 이용하여 서명을 하는 방식을 취하고, B-EKE는 사용자가 서버로부터 받은 임의의 난수 g^x에 pwd를 지수승한 g^{x*pwd}을 서버에게 보냄으로써 인증한다. AuthA는 사용자가 패스워드와 사용자 식별자 값을 해쉬 함수에 적용시킨 결과 $H(ID\|B\|pwd)$를 서버로부터 받은 임의의 난수 g^b에 지수승하여 $g^{b*H(ID\|B\|pwd)}$을 보냄으로써 인증한다. 이러한 인증 방법을 사용함으로써 수반되는 부수적인 전송횟수나, 지수승 계산 비용에 대하여 기존의 프로토콜들은 효율성 개선의 여지가 있다.

본 논문에서는 보안 요구사항들을 만족하면서도 일회용 패스워드를 사용하여 인증 단계의 효율성을 증대시킨 OTP-EKE 프로토콜을 제안한다.

■ 사용자 ID와 패스워드를 설정하는 초기 단계

사용자가 서버로부터 서비스를 받기 위하여 ID와 패스워드 확인자 $H^n(pwd)$를 안전한 채널을 통해 서버에게 전송한다. 서버는 사용자에게 받은 ID와 $H^n(pwd)$를 패스워드 디렉토리 내에 저장하고, 서버를 인증할 수 있는 공개 패스워드(public password)[7] $H(g^s)$을 안전한 채널을 통해 사용자에게 전송한다. 여기서 $H^n(pwd)$는 pwd에 일방향 해쉬함수를 n번 적용해 얻은 값으로 초기 통신에서 사용하고, i 번째 통신에서는 일회용 패스워드[6] 방식처럼 $H^{n-i+1}(pwd)$를 패스워드의 확인자로 사용한다. 설정한 패스워드 확인자를 $n-1$번 사용한 후에는 새로운 패스워드를 설정한다. 또한 공개 패스워드 $H(g^s)$는 서버의 장기 공개키(long-term public key) g^s의 해쉬된 값으로서 사용자가 기억하거나 소유하고 있을 만큼 충분히 길이가 짧은 값(60~80 bit)이다. 이것은 기밀성(security)을 보장할 필요가 없지만, 무결성(integrity)은 보장되어야 한다[7].

■ 인증 및 세션키 설정 단계

본 논문에서 제안하는 프로토콜 OTP-EKE에서 서버와의 패스워드 설정 후 i 번째 통신에 대한 단계별 수행 과정은 다음과 같으며 그림 1에 요약되어 있다.

① A는 $a\in_R[1,q-1]$를 선택해서 g^a을 계산하고 $H^{n-i+1}(pwd)$로 암호화하여 ID와 함께 B에게 전송한다.

② B는 $b\in_R[1,q-1]$를 선택해서 g^b을 계산하고, A에게 받은 값을 이용해 $K=g^{ab}, K^*=g^{as}$을 계산한다. 또한 키 확인 메시지 $H(K\|K^*)$을 생성하여, $H^{n-i+1}(pwd)$로 암호화한 g^b을 g^s와 함께 A에게 전송한다. 사용자의 사용 환경에 따라 g^s은 저장될 수도 있는 값이고, 저장할 경우 서버가 매번 프로토콜 실행할 때마다 전송하지 않아도 된다.

③ A는 B로부터 전송 받은 값을 $H^{n-i+1}(pwd)$로 복호화하고, g^s에 일방향 해쉬함수를 적용하여 자신이 가지고 있는 공개 패스워드 $H(g^s)$값과 비교해서 같은지를 확인하고, 같지 않으면 프로토콜 세션을 종료한다. 여기서, B가 서버의 장기 비밀키 s를 알고 있음은 B가 K^*를 사용하여 만든 키 확인 메시지를 통해 검증된다. A는 $a\in_R[1,q-1]$를 선택해서 g^a와 $K=g^{ab}$, $K^*=g^{as}$을 계산하고 $H(K\|K^*)$값을 확인한 후, 다음 세션을 위한 패스워드 확인자 $H^{n-i}(pwd)$와 세션키 확인 메시지 $H(K)$를 K^*로 암호화해서 B에게 전송한다. 여기서 $K=g^{ab}$는 세션키이고, $K^*=g^{as}$은 사용자의 다음 번 패스워드 확인자를 안전하게 전송하기 위해서 사용되는 암호화키이다. K^*는 사용자와 서버만이 만들 수 있는 값이기 때문에 공격자는 이를 만들지 못하므로 K^*를 다음 번 사용자의 패스워드 확인자를 암호화하여 보내는데 사용된다.

④ B는 A로부터 전송 받은 $H(K)$를 확인하여 A와 세션키 K를 공유했음을 확인한 후, 전송 받은 확인자 값에 일방향 해쉬함수를 적용하여 $H(H^{n-i}(pwd))=H^{n-i+1}(pwd)$인지를 확인한다. 등식이 성립하면, 서버는 사용자를 인증하고, A의 패스워드 확인자를 $H^{n-i}(pwd)$로 교체해서 저장한다.

단계	A	메시지	B
1	$a\in_R[1,q-1]$	$ID, E_{H^{n-i+1}(pwd)}(g^a)$ →	$b\in_R[1,q-1]$
2	$H(g^s)$확인 $K=g^{ab}$ $K^*=g^{as}$	← $E_{H^{n-i+1}(pwd)}(g^b), g^s,$ $H(K\|K^*)$	$K=g^{ab}$ $K^*=g^{as}$
3		$E_{K^*}(H^{n-i}(pwd), H(K))$ →	$H(H^{n-i}(pwd))$ $=H^{n-i+1}(pwd)$

그림 1 OTP-EKE 프로토콜

4. 안전성 및 효율성 분석

이 절에서는 OTP-EKE의 안전성을 분석하고, 효율성을 비교 분석한다.

■ 안전성 분석

패스워드 기반 키 교환 프로토콜은 적은 비트 길이의 패스워드를 이용하여 세션키를 공유하기 때문에, 패스워드에 대한 온라인 추측공격과 서버의 패스워드 확인자 노출시의 사전공격은 막을 수 없다. 제안한 프로토콜은 이를 제외한 보안요구사항을 다음과 같이 만족한다.

① 도청 공격 : OTP-EKE에서 전송되는 메시지들이 추측 불가능한 임의의 난수들이면서 모두 암호화되어 전송되기 때문에 안전하다.

② 재전송 공격 : 프로토콜 단계별로 같은 메시지가 연속적으로 전송되지 않고, 매 세션마다 새로 생성되는 랜덤 수와 패스워드 확인자를 사용하기 때문에 안전하다.

③ 중간 침입자 공격 : OTP-EKE에서 g^a, g^b은 패스워드 확인

자로 암호화하여 전송되기 때문에 알아낼 수 없고, 공격자가 패스워드 확인자를 알고 있다면 g^a, g^b은 알아내어 g^a, g^b로 대치할 수 있지만 사용자가 $H(g^s)$값을 알고 있으므로 g^s은 g^s로 대치할 수 없다. 또한 Diffie-Hellman 문제의 어려움에[11] 근거하여 g^a, g^s로부터 $K^* = g^{as}$을 알아낼 수 없으므로 이 공격에 대해 안전하다.

④ 오프라인 패스워드 추측 공격 : 패스워드 확인자로 암호화된 메시지는 추정 가능문(verifiable-text)이나 기지 평문(known-plaintext)이 아니기 때문에 사전 공격을 수행해서 올바른 패스워드를 알아낸다는 것은 불가능하다.

⑤ Dennig-Sacco 공격 : OTP-EKE의 세션키 K는 사용자의 패스워드 정보를 포함하지 않으므로 세션키가 누출되더라도 사용자의 패스워드에 관한 정보를 알 수 없다. 또한 매번 랜덤하게 생성되는 난수들을 이용하여 세션키를 만들기 때문에 세션키가 누출되었다고 할지라도 그 이후의 세션키에 관한 정보는 알 수 없다.

⑥ Perfect forward secrecy: Diffie-Hellman 문제의 어려움에 근거하여 OTP-EKE에서 장기간 키로 사용되는 사용자의 패스워드가 노출되어도 과거에 사용되었던 세션키의 값들은 알 수 없다.

■ 효율성 비교 분석

OTP-EKE는 온라인 상의 메시지 전송 회수가 3회로 기존 프로토콜들에 비하여 가장 적다. 또한 사용자와 서버가 병렬적으로 수행하는 모듈러 지수승 계산 횟수를 비교해 보면, OTP-EKE는 3회로서 가장 효율적이며, 그림 1의 단계 1에서 수행하는 g^a, g^b 계산을 온라인 통신 전에 미리 계산해 놓으면, 병렬 지수승 계산은 2회로서 온라인의 실행시간을 줄일 수 있다. 난수 생성에서도 서버와 사용자가 세션키를 만들기 위해서 난수 생성을 한번씩만 하면 된다. 표 1은 확인자 기반의 프로토콜들과 OTP-EKE를 라운드 수, 지수승 계산 횟수, 난수 생성 횟수 등으로 나누어 분석하고 효율성을 비교한 결과이다.

표 1 확인자 기반방식 프로토콜의 비교분석

	메시지 전송 횟수	지수승 계산 횟수			난수 생성 횟수	
		사용자	서 버	Parallel	사용자	서 버
A-EKE[2]	5	4	4	6	1	1
B-SPEKE[8]	4	3	4	6	1	2
SRP[12]	4	3	3	4	1	1
B-EKE[8]	4	3	4	4	1	2
SNAPI-X[10]	5	5	4	7	2	3
AuthA[1]	3	4	3	6	1	1
PAK-X[4]	3	4	4	8	1	2
AMP[9]	4	2	4	5	1	1
OTP-EKE	3	3	3	3	1	1

5. 결론

본 논문에서는 일회용 패스워드 방식을 적용한 확인자 방식의 새로운 패스워드 기반 키 교환 프로토콜 OTP-EKE를 제안하였다. OTP-EKE는 세션키 공유 시에 메시지들을 패스워드 확인자로 암호화하여 보내는 EKE부류의 확인자 기반 프로토콜로서 필수적인 보안 요구 사항에 맞추어 설계되었다. 특히 OTP-EKE는 사용자 패스워드 증명 단계에서 일회용 패스워드 방식을 적용함으로써 기존의 방식들보다 모듈러 지수승 계산을 적게 하여 효율성을 높였으며, 메시지 전송회수를 3회로 최소화시켰다. 본 논문에서 제안한 OTP-EKE는 안전도를 저하시키지 않으면서, 기존 확인자 기반 방식의 프로토콜들보다 개선된 효율성을 가진다.

6.참고 문헌

[1] M.Bellare, D.Jablon, H.Krawczyk, P.MacKenzie, P.Rogaway, R.Swaminathan, T.Wu," Proposal for P1363 Study group on password-based authenticated -Key-Exchange Methods," 2000.

[2] S.Bellovin, M.Merritt, "Augmented encrypted key exchange: a password-based protocol secure against dictionary attacks and password-file compromise," ACM Conference on Computer and Communications Security, 1993.

[3] S.Bellovin, M.Merrit, "Encrypted key exchange : password-based protocols secure against dictionary attacks," IEEE Symposium on Research in Security and Privacy, 1992.

[4] V.Boyko, P.MacKenzie, S.Patel, "Provably secure password authenticated key exchange using Diffie-Hellman," Eurocrypt 00, 2000.

[5] M.Bellare, P.Rogaway, "The AuthA protocol for password-based authenticated key exchange," 2000, available from http://www.cs.ucdavis.edu/rogaway/papers/autha.ps.

[6] N.Haller, "The S/KEY one-time password system," RFC 1760, 1995.

[7] S. Halevi, H. Krawczyk, "Public-key cryptography and password protocols," ACM Transactions on Information and System Security (TISSEC), Vol. 2, 1999.

[8] D.Jablon, "Extended password key exchange protocols," WETICE Workshop on Enterprise Security, 1997.

[9] T.Kwon, "Authentication and key agreement via memorable password," NDSS 2001 Symposium Conference Proceedings, 2001.

[10] P.MacKenzie, S.Patel, R.Swaminathan, "Password-authenticated key exchange based on RSA," ASIACRYPT, 2000.

[11] D.R.Stinson, *Cryptography Theory and Practice*, CRC.

[12] T.Wu, "Secure remote password protocol," NDSS, 1998.