패스워드 기반의 독립적 인증 및 키 교환 프로토콜

김 지 영°, 정 경 숙, 정 태 충
경희대학교 전자계산공학과
redciel@iislab.kyunghee.ac.kr, jungks@iislab.kyunghee.ac.kr, tcchung@khu.ac.kr

Password-based Independent authentication and Key Exchange protocol

Ji-young Kim°, Kyoung-sook Jung, Tae-choong Chung
Dept. of Computer Engineering, KyungHee University

요 약

본 논문은 신뢰할 수 없는 네트워크를 통해서 사용자를 인증하거나 키를 교환하는 것에 적합한 패스워드 기반 프로토콜을 제안한다. 기존의 패스워드 기반의 프로토콜들은 클라이언트와 서버 사이에 인증기관(CA)을 통하여 사용자를 인증하는 반면, 본 논문에서는 사용자와 서버가 독립적으로 키 교환 및 인증을 하는 패스워드 기반 프로토콜을 제안한다. 충분하지 않은 패스워드의 랜덤성과 짧은 길이로 인하여 패스워드를 사용해 인증 및 키 교환을 하는 것은 많은 주의를 요한다. 그러므로 Diffie-Hellman 키교환 방식에 기반한 SRP 프로토콜과 ECDSA의 서명 기법을 적용하여 안전성이 높은 프로토콜을 제안한다.

1. 서 론

지식 정보화 사회에서 인터넷이 발달됨에 따라 온라인상에 노출되는 정보들에 대한 불법적인 위·변조 및 신분 위장 등 각종 위협이 예상되고 있다. 그러므로 인터넷상에서 사용자와 정보 제공자간의 정보보호를 위해 상호간의 인증(Authentication)이 중요한 문제가 되었다.

정보 유통시 안전성과 신뢰성 확보를 위해 각종 분야에 공개키 암호기술을 적용한 인증서(certification) 기반의 PKI(공개키 기반 구조:Public Key Infrastructure)가 현재 가장 보편화되어 있는 방법이다. PKI에서는 사용자의 신상정보와 공개키를 확인할 수 있도록 제 3자인 인증기관(CA : certificate authority)으로부터 인증서를 발급받는다. 그러나 잦은 인증서 발급으로 통화량의 증가와 비용 및 시간의 소모, 키 관리 등 복잡한 문제가 발생하고 있다. 따라서 사용자간에 실질적인 통신 및 전자상거래시 제3의 신뢰기관과의 접촉없이 독립적으로 안전한 사용자 인증 및 키 분배가 가능한 시스템에 대한 연구가 필요하게 되었다. 이러한 목적을 갖는 프로토콜은 여러 종류가 있지만 패스워드 이용하는 인증 프로토콜은 사용자들이 자신이 설정한 패스워드를 이용해 인증할 수 있는 방법으로 가장 효율적이다. 그러나 패스워드의 충분치 못한 짧은 길이와 랜덤성으로 인해 인증 및 키 교환 프로토콜의 설계에 패스워드를 사용하는 것은 많은 주의를 요한다.[1] 그러므로 패스워드 기반 프로토콜들은 다양한 공격들로부터 대응할 수 있도록 구성되어야 한다. 본 논문에서 제안하는 프로토콜은 기존의 제 3의 신뢰기관에 접근하지 않고 Client와 Server간에 독립적으로 사용자 인증 및 키 교환이 가능한 패스워드 프로토콜로 ECDSA 서명기법을 이용하여 신뢰할 수 없는 네트워크를 통해서도 사용자를 인증하거나 키를 교환할 때 안전성을 향상시키는 것에 목적이 있다. 본 논문의 2장에서는 패스워드 기반 인증 프로토콜 SRP와 ECDSA의 절차를 간단히 설명하고, 3장에서는 ECDSA기법을 이용한 패스워드 기반 인증 및 키 교환 프로토콜을 제안한다. 4장에서는 제안 프로토콜의 특징과 안전성을 설명하고, 5장에서는 결론 및 향후 연구 방안에 대하여 논한다.

2. 관련 연구

2. 1 패스워드 기반 프로토콜

패스워드 기반의 프로토콜 중 대표적인 것들로 EKE[2], A-EKE, SPEKE[4], SRP, AMP등이 있다. 그 중, EKE(Encrypted Key Exchange)는 불충분한 정보로 추측된 패스워드를 검증하려는 능동 공격자로부터의 사전 공격(dictionary attack)에 강한 프로토콜이다.[2,3] 그러나 EKE는 클라이언트와 호스트가 동일한 비밀 패스워드 혹은 그것들의 해쉬 값으로 접근하는 단점이 있다. 다른 방법으로는 Thomas Wu의 SRP(Secure Remote Protocol)으로 패스워드 파일을 비대칭으로 저장해 파일 노출시 패스워드가 직접 노출되지 않게 하고, 네트워크 상에서 어떠한 패스워드의 정보를 유출시키지 않는 영지식(zero-Knowledge)을 지향하는 프로토콜이다.[5]

2. 2 SRP

SRP(Secure Remote Password) 프로토콜은 Diffie-Hellman 키교환 방식에 기반한 프로토콜로 두 참여자의 키 교환 설정단계에서 이산대수 문제를 이용하여 구성하고, 두 참여자간의 상호인증은 해쉬함수를 이용하여 구성된다. 일방향 인증을 수행하기 위해 상호 연동적으로 다음과 같이 프로토콜을 실행한다.

● 시스템 설정

n이 큰 소수(large prime number)이라고 할 때, 소수가 되는

$$q=2*n+1 \ , \ p=k*q+1$$

을 선택한다.(k는 짝수) $g \in_N Z_p^*$는 order q의 원소라고 하면,

$$g^q \equiv 1 \ mod \ p$$

와 같은 형태를 갖게 된다. 여기에서 밑수(base) g의 이산대수를 계산하는 것은 불가능하다고 가정한다.

● 실행 단계

① 우선 클라이언트가 pwd를 설정한다.
② salt값 s를 선택하고 $x = Hash(s,pwd)$, $v = g^x \bmod p$를 계산한다.
③ 클라이언트는 사전지식으로 (pwd, q, p, g)를 갖고, 서버에게 (s, v)를 전달한다.
④ 클라이언트는 난수 $a \in_N Z_q$를 생성하고, 자신의 임시 공개키 $A=g^a$를 계산한 다음 서버에게 전송한다.
⑤ 서버에서는 난수 $b,u \in_N Z_q$를 생성하여 자신의 임시 공개키 $B = v + g^b$를 계산해 (B, u)를 클라이언트에 전송한다.
⑥ 클라이언트와 서버는 각각 지니고 있는 값을 이용해 공통 지수값 S를 계산하고, $SK=Hash(S)$를 계산해 상호 동의된 세션키를 설정한다.
⑦ 클라이언트는 서버에게 정확한 S를 지니고 있음을 증명하는 M을 구성해 전달하면 서버가 검증을 통해 확인한다.

2. 3 ECDSA

ECDSA(Elliptic curve DSA)는 DSA를 타원곡선 알고리즘으로 옮긴 것으로 ANSI X9.62[7]로 표준화되었다. 기존의 DSA알고리즘보다는 ECDSA를 사용하는 것이 키 bit당 암호화 정도가 강하다고 할 수 있다.[8]

● 실행 단계

1) ECDSA 키 생성

① Z_p에서 정의된 타원곡선 E를 선택한다.
② 위수가 n인 점 $P \in E(Z_p)$를 선택한다.
③ 구간 $[2,n-2]$에서 난수 d를 선택한다.
④ $Q = dP$를 계산한다.
⑤ 사용자의 공개키(Public Key)는 (E,P,n,Q)이며 사용자의 비밀키(Secret Key)는 d이다.

2) ECDSA 서명 생성

① 구간 $[1,n-1]$에서 난수 a를 선택한다.
② $aP=(x_1,y_1)$과 $r=x_1 \bmod n$을 계산한다.(x_1은 정수)
③ $r=0$이면, ①단계로 되돌아간다.
④ $a^{-1} \bmod n$을 계산한다.
⑤ $s=a^{-1}(h(m)+dr) \bmod n$을 계산한다. ($h$: message m의 SHA-1 해쉬 알고리즘)
⑥ $s=0$이면, ①단계로 되돌아간다.
⑦ 메시지 m에 대한 서명은 (r, s)이다.

3) ECDSA 서명 검증

① 사용자의 서명 (r, s)를 검증하기 위해서 서버는 사용자의 인증된 공개키 (E,P,n,Q)를 얻는다.
② r과 s가 $[1,n-1]$에 있는 지 확인한다.
③ $w=s^{-1} \bmod n$과 $h(m)$을 계산한다.
④ $u_1=h(m)w \bmod n$와 $u_2=rw \bmod n$을 계산한다.
⑤ $u_1P+u_2Q=(x_0,y_0)$를 계산 $v=x_0 \bmod n$
⑥ 만약 $v=r$이면 올바른 서명이다.

3. 제안 프로토콜

기존의 SRP는 키 교환 설정 단계가 이산대수문제에 근거하므로 타원곡선을 적용할 수 있다. 제안한 프로토콜은 <그림 2>와 같이 안전성 향상 및 프로토콜의 간략화를 위해 일방향 최적화된 SRP에 ECDSA를 적용하였다.

● 설정

사전 공격을 보다 어렵게 하기 위하여 본 논문에서는 salt 값을 추가해 pwd와 연접시켜 일방향 함수에 입력한다.[6]

$$x = Hash(salt, pwd)$$

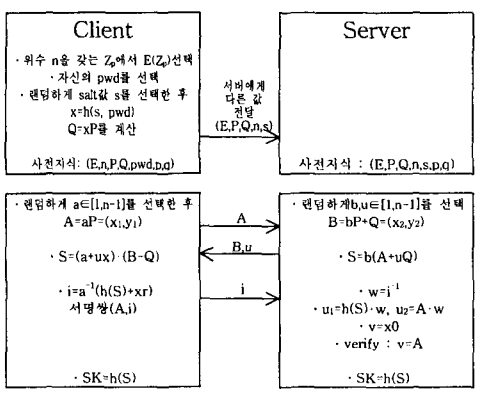
타원곡선 E를 단순화하기 위해 $K = F_p = Z_p$ (p:소수,p개의 원소를 갖는 유한체)로, 타원곡선는 $y^2 = x^3 + ax + b$ $(a,b \in Z_p)(4a^3+27b^2 \neq 0 (mod\ p))$와 무한원점$(0)$을 말하기로 한다.

n은 160비트 이상의 크기를 갖는 소수, 타원곡선 $E(Z_p)$는 $y^2=x^3+ax+b(a,b \in Z_p)$의 방정식을 만족하는 Z_p상의 점들과 무한점으로 이루어진 집합을 의미한다.

① Z_p위에 정의된 타원곡선 $E(Z_p)$를 선택한다. ($E(Z_p)$는 큰 소수 n에 의해 나누어져야 한다.)
② 위수가 n인 $P \in E(Z_p)$를 선택한다.
③ 안전한 일방향 해쉬함수 h를 선택한다.
④ 클라이언트가 pwd를 설정한다.
⑤ salt값 s를 선택하고 $x = h(s,pwd)$를 계산하고 점 $Q = xP$를 계산한다.
⑥ 클라이언트는 사전지식으로 (E,P,Q,pwd,q,p) 를 갖고, 서버에게 (E,P,n,Q,s)를 전달한다.

●실행 단계

① 구간 $[1,n-1]$에서 임의의 난수 a를 선택하고, $A=aP=(x_1,y_1)$를 server로 전송한다.
② 난수 $b,u \in [1,n-1]$를 선택해, $B=bP=(x_2,y_2)$를 계산하고, (B,u)를 Client로 전송한다.
③ Client와 서버는 각각 공통 값 S 계산
④ $i=a^{-1}(h(S)+xr)$를 계산한다. 그리하여 서명쌍은 (A,i)가 되고, Server에 i값을 전달한다.
⑤ Server에서 $w=i^{-1} \bmod n$을 계산하여, $u_1=h(S)w \bmod n$, $u_2=Aw \bmod n$를 계산한다.
⑥ $u_1P+u_2Q=(x_0,y_0)$, $v=x_0 \bmod n$을 계산해 $v=A$이면 올바른 서명이다.
⑦ Client와 Server는 세션키 $SK=h(S)$를 생성한다.



<그림 1> 제안 프로토콜

4. 제안 프로토콜 분석

4.1 제안 프로토콜의 특징

네트워크 상에서의 통신 회수는 네트워크 자원의 효율

성과 네트워크상의 지연(delay) 등을 고려할 때 적을수록 장점을 갖는다. 그러나 통신 회수가 줄어도 보안의 안전도는 변함이 없어야 한다.

제안 프로토콜은 사용자의 서명을 이용함으로써 사용자의 부인 방지를 제공하고, 인증에 사용되는 키 쌍 (A,i)는 두 통신자 사이의 키 교환(A와 B)에 의해 생성된다. <그림 1>을 이용해 설명하면, A와 i값은 사용자에 의해 생성되는 서명 값으로 사용자가 자신의 비밀키 $x=h(s, pwd)$를 이용해 서명 값을 생성하고, 서버는 사용자의 공개키(E,P,Q,n,s)를 이용해 서명을 검증하여 인증하게 됨으로 생성한 세션키 S에 대한 사용자의 부인을 방지할 수 있다. 그리고, 사용자의 인증에 사용되는 키 쌍은 매 세션마다 새롭게 생성되는 값이며, 세션키도 매 세션마다 생성된다.

4.2 제안 프로토콜의 안전성

● 사전공격

사전공격(dictionary attack)은 공격자가 사용자의 패스워드를 추측하여 실제 메시지에서 드러나는 값에 대입하여 결과를 비교해 실제 패스워드를 찾는 방법이다. 제안 프로토콜에서는 패스워드에 salt값을 추가하여 일방향 해쉬함수에 적용하므로 사전 공격을 차단할 수 있다.

● Replay attack

공격자가 사용자의 메시지를 재전송하여 이미 정상적인 사용자에 의해 생성된 이전키를 다시 생성하기 위함이다. 이 공격방법은 사용자와 서버간에 항상 임의의 값인 a와 b가 매 세션마다 새로 생성됨으로 불가능하다.

● Perfect Forward Secrecy

현재의 세션키 정보가 알려져도 이전키를 알 수 없는 것으로 생성되는 세션키 값이 항상 임의의 값으로 생성되어 Perfect Forward Secrecy를 제공한다.

● Denning-Sacco attack

이전키를 알아내 패스워드를 알아내는 방법이다. 이 공격은 세션키가 임의의 값으로 매 세션마다 생성되고, 만약 S값이 알려지더라도 패스워드 정보를 갖고 있지 않기 때문에 불가능하다.

4.3 성능분석

프로토콜 수행과정에서 수행속도와 밀접한 관련이 있는 연산은 해쉬함수 연산과 지수연산의 회수이다. 제안 프로토콜에서 해쉬함수 연산은 클라이언트에서 2번, 서버에서 1번 수행되며, 지수연산은 클라이언트 2번, 서버에서 2번 이루어진다. 연산이 이루어지는 부분은 다음과 같다.

	해쉬함수연산	지수연산
Client	$x=h(s, pwd)$	$A=aP=(x_1,y_1)$
	$i=a^{-1}(h(S)+xr)$	$S=(a+ux)\cdot(B-Q)$
Server	$u_1=h(S)\cdot w$	$B=bP+Q=(x_2,y_2)$
		$S=b(A+uQ)$

<표 1> 해쉬함수와 지수 연산 회수

구분	프로토콜	Pass	암호화		Hash 함수		지수연산		랜덤생성	
			Client	Server	Client	Server	Client	Server	Client	Server
공개키	A-EKE	5	RSA서명		x	x	2	2	1	1
	SNAPI-K	5	RSA		4	3	3	4	2	2
	AMP	2	DSS		3	2	2	2	1	1
	AKEECC	4	ECDSA		3	3	2	2	1	1
	A-EKE	5	3	3	1	1	2	2	1	1
DH	B-SPEKE	4	x	x	1	1	3	4	1	2
	AMP	4	x	x	4	4	2	2	1	1
	SRP	4	x	x	3	2	3	3	1	1
	제안프로토콜	3	ECDSA		2	1	2	2	1	1

<표 2> pass와 계산량 비교

5. 결론 및 향후 연구 방향

본 논문에서는 패스워드의 충분치 않은 랜덤성과 짧은 길이로 인한 패스워드 기반 인증 및 키 교환을 보완하고 안전성을 높이는 프로토콜을 제안하였다. 이 제안 프로토콜은 기존의 패스워드 프로토콜이 클라이언트와 서버 사이에 인증기관(CA)을 통하여 인증하던 방식과 달리 사용자와 서버가 독립적으로 키 교환 및 인증을 하는 패스워드 기반 프로토콜을 하도록 Diffie-Hellman 키교환 방식에 기반한 SRP 프로토콜을 사용하고, 안전성과 효율을 높이기 위해 ECDSA의 서명 기법을 적용하였다.

6. 참고문헌

[1] R. Anderson and T. Lomas, "Fortifying Key negotiation schemes with poorly chosen passwords", Electronics Letters, 1994, Vol. 30, No. 13,

[2] S. Bellovin and M. Merritt, "Encrypted Key exchange: password based protocols secure against dictionary attacks", IEEE Comp. Society Symp. on Research in Security and Privacy, 1992. page.72-81.

[3] S. Bellovin and M. Merritt, "Augmented Encrypted Key Exchange", in Proceedings of the First ACM Conference on computer and communications Security, page. 244-250, 1993

[4] D. Jablon, "Strong Password-only Authenticated Key Exchange", Computer Comm Review, ACM SIGCOMM, vol.26, no. 5, page. 5-26, 1996

[5] Thomas Wu, "The Secure Remote Password Protocol", Internet Society Symp. Network and Distributed Systems Security Symposium, 1998, page. 97-111.

[6] S. Even, O. Goldreich, A. Lempel "A Randomized Protocol for Signing Contracts", Communications of the ACM, 28, 1985, page. 637-647.

[7] ANSI X9.62, The elliptic curve digital signature algorithm(ECDSA), *draft standard*, 1997.

[8] Miller, V., Uses of elliptic curves in cryptography, *Advances in Cryptology, CRYPTO 85 - Lecture Notes in Computer Science*, Volume 218, Springer-Verlag, pages 417-426, 1986