패스워드 인증 프로토콜 동향

Trend on Password-Based Authentication Protocols

김영수(Y.S. Kim) 능동프로토콜연구팀 연구원
나중찬(J.C. Na) 능동프로토콜연구팀 선임연구원, 팀장
손승원(S.W. Sohn) 네트워크보안연구부 책임연구원, 부장

사용자들이 일반적으로 사용하는 패스워드의 약한 안전성으로 인한 보안 사고는 매우 보편화되었고, 그 피해 또한 상당하다. 인터넷이 점차적으로 공격자들에게 개방되고 있는 현재 환경에서, 약한 패스워드 시스템들에 대한 보안 사고는 그 수를 헤아릴 수 없을 만큼 증가하고 있는 실정이다. 본 고는 기존의 패스워드-인증 관련 보안 취약점들의 심각성을 지적하고, 최근 이슈가 되고 있는 패스워드-기반 인증 및 키 교환 프로토콜들의 특성과 현재까지의 연구 동향에 대하여 고찰하고 특히, SRP(Secure Remote Password) 프로토콜에 대하여 자세히 기술한다.

I. 배 경

인터넷상의 데이터나 시스템 가치가 높아짐에 따라, 최근 몇 년 동안 악의적 목적을 가진 공격자의 능력은 급속히 향상되었다. 사용자들이 일반적으로 사용하는 패스워드의 약한 안전성으로 인한 보안 사고는 매우 보편화되었고, 그 피해 또한 상당하다. 인터넷이 점차적으로 공격자들에게 개방되고 있는 현재 환경에서, 약한 패스워드 시스템들에 대한 보안 사고는 그 수를 헤아릴 수 없을 만큼 증가하고 있는 실정이다. Kerberos[1]가 안전한 분산 인증 시스템으로 인정 받기 시작했으나, Kerberos V4와 사전-인증 없는 V5의 초기 티켓-허가 프로토콜들이 안전하지 않다는 것이 밝혀져 인증 서버에 네트워크 접속하는 사람이면 모두 마스터 패스워드 리스트를 대상으로 패스워드-크래킹을 행할 수 있다. 인터넷 프로바이더들이 데이터를 암호화하는 프로토콜과 소프트웨어를 제공하거나, 패스워드를 클리어한 상태로 인터넷상에 전송하지 않아도 되는 대체 시스템을 제공해야 한다고 주장하는 전문가들도 있다. 본 고는 기존의 패스워드-인증 관련 보안 취약점들의 심각성을 지적하고, 최근 이슈가 되고 있는 패스워드-기반 인증 및 키 교환 프로토콜들의 특성과 현재까지의 연구 동향에 대하여 고찰한다.

II. 패스워드-기반 프로토콜의 개념

일반적으로 신분 인증(human authentication)은 다음과 같이 크게 세 가지로 구분할 수 있다.

- 사용자의 신체적 특징(음성 식별, 망막 스캐닝)
- 사용자가 가진 것(신분증, 스마트카드)
- 사용자가 기억하는 것(패스워드)

여기에서는 직접 패스워드 인증으로 알려진 세 번째 경우만을 다룬다. 이 메커니즘에서는 사용자의 패스워드만이 유일한 비밀 정보이며 또한 클라이언트와 서버 사이의 네트워크는 도청 등에 의하여 공

격 받기 쉽고, 키 서버나 중재자와 같은 신뢰기관(Trusted Third Party: TTP) 없이 오직 양측 사용자만이 인증 프로토콜에 관여한다고 가정한다. 이 같은 프로토콜은 패스워드 만을 필요로 하므로 사용이 간편하고, 바이오 메트릭이나 토큰-기반 방식에 비해 상대적으로 저렴하므로 다양한 분야에 적용될 수 있다. 예를 들어, 유닉스 텔넷 프로그램 같은 다중 사용자 시스템의 원거리 로그인 인증이나 웹 사이트의 사용자-기반 접근 제어, X-터미널 또는 네트워크 컴퓨터 인증, Novell사의 Netware 같은 LAN상의 네트워크 서비스 인증, 그 밖에도 공중전화, 휴대전화, 현금인출기, 셋탑 박스 등 그 응용 범위가 매우 크다.

III. 패스워드 인증의 3가지 이슈

패스워드를 이용한 인증을 성공적으로 수행하기 위해서는 사전 공격(dictionary attack)에 대하여 안전해야 하고, 평문 동등성(plaintext equivalence)을 피하고 전향적 안전성(forward secrecy)을 가져야 한다.

1. 사전 공격

사전 공격은 일종의 패스워드 추측 공격(password guessing attack)으로 사용자들이 일반적으로 자신이 기억하기 쉬운 패스워드를 사용한다는 사실에 기초한 공격 방법이다. 사전 공격은 패스워드-데이터베이스 기반 사전 공격과 네트워크 기반 사전 공격으로 크게 나눌 수 있다.

가. 패스워드-데이터베이스 기반 사전 공격

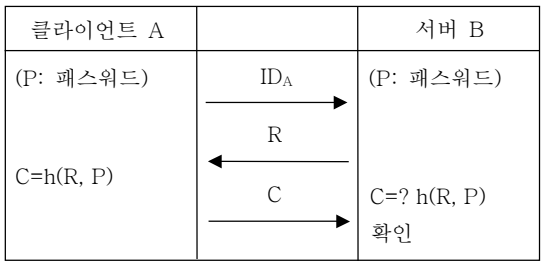
이 공격 방법은 유닉스를 예로 들 수 있다. /etc/passwd 파일의 해싱된 패스워드 엔트리는 64비트로서 하나의 56비트 키로부터 생성되므로, 이를 알아내기 위해서는 2^{56}회의 시도를 해야 한다. 그러나, 키는 단순한 패스워드의 함수이므로, 공격자는 보통 자주 사용되는 패스워드 사전(dictionary)을 만들어 이를 먼저 검색해 본다. 백만 개의 단어로 된 사전의 경우, 무작위로 행하는 전사적(brute-force) 공격 시간이 전체 사전검색 시간의 10^{13}배 정도 된다.

나. 네트워크 기반 사전 공격

간단한 Challenge-Response 프로토콜을 예로 들어 보자.

(그림 1)은 클라이언트 A가 자신의 패스워드 P를 통해 서버 B에 인증 받기 위한 간단한 Challenge-Response 프로토콜이다. 클라이언트 A는 자신의 개인 식별 정보인 ID_A를 서버 B에게 전송하고, 서버 B는 랜덤수 R을 생성하여 클라이언트 A에게 전송한다. R을 받은 클라이언트 A는 R과 자신의 패스워드 P를 함께 해싱한 값 C(=h(R, P), 여기서 h()는 해시 함수)를 서버 B에게 보낸다. 서버 B는 받은 C 값과 자신이 계산한 h(R, P) 값이 같은지 여부를 확인하여 같으면 인증에 성공하게 된다.

그러나, 아래 동작 과정에서 공격자가 A와 B 사이에 전송되는 모든 정보를 기록할 경우를 생각해 보자. 획득한 R과 추측한 P'로 h(R, P) =? h(R, P') 인지 확인하고, 이것이 성공하게 되면 공격자는 P를 알게 되며, 클라이언트 A로 위장할 수 있게 된다. 본 예는 휴대전화 네트워크, 케이블 접속용 셋탑 박스, 공중 인터넷 접속 포인트 등에 적용이 가능하다.



(그림 1) Challenge-Response 프로토콜

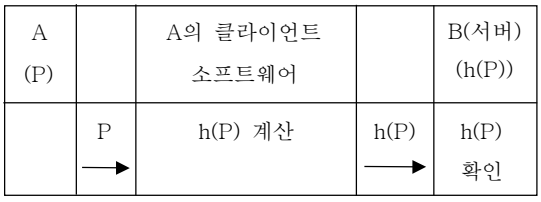
2. 평문 동등성

어떤 데이터가 실제 패스워드를 통한 접속과 동일한 수준의 접속을 위해 사용된다면, 이러한 데이

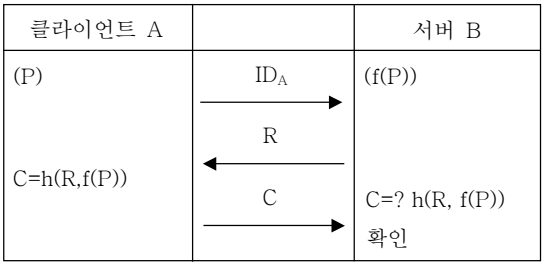
터를 특정 패스워드에 대하여 “평문 동등(plain text equivalent)하다”라고 정의한다. (그림 2)와 (그림 3)은 이러한 개념을 보여주는 좋은 예들이다.

(그림 2)는 단순한 원격 로그인 프로토콜이다. 사용자 A가 패스워드 P를 입력하면, A의 클라이언트 소프트웨어가 표준 유닉스 해시 함수를 이용하여 h(P)를 계산하여 서버 B에게 전송하고 h(P)를 미리 저장하고 있던 서버 B는 수신한 h(P) 값을 확인한다.

(그림 3)에서 클라이언트 A는 자신의 패스워드 P를, 서버 B는 패스워드 P의 일방향 함수 값(f(P))을 저장하고 있다. 클라이언트 A는 자신의 개인 식별 정보인 ID_A를 서버 B에게 전송하고, 서버 B는 첼린지 스트링인 랜덤수 R을 생성하여 클라이언트 A에게 전송한다. R을 받은 클라이언트 A는 R과 패스워드 P의 일방향 함수 값을 함께 해싱한 값 C(=h(R, f(P)), 여기서 h()는 해시 함수)를 서버 B에게 보낸다. 서버 B는 받은 C 값과 자신이 계산한 h(R, f(P)) 값이 같은지 여부를 확인하여 같으면 인증에 성공하게 된다. 여기서 P가 서버 B에 저장되지 않지만, 공격자는 서버의 패스워드 데이터베이스로부터 f(P)를 찾아서 A로 위장할 수가 있는데, 이는 클라이언트측의 계산이 언제나 f(P) 값을 사용하고 P에만 종속되기 때문이다.



(그림 2) 원격 로그인 프로토콜



(그림 3) 일방향 함수 f()를 이용한 Challenge-Response 프로토콜

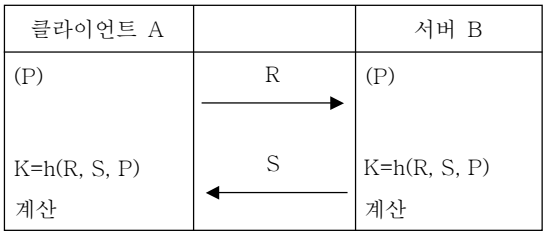
패스워드-인증 시스템에서 평문 동등성은 실제로 해결하기 매우 어려운 문제이다. 양측이 동일한 비밀을 가지고 있다고 가정하면, 프로토콜을 설계하기가 매우 쉽다. 양측은 분배된 비밀과 교환된 메시지들을 이용하여 복잡하지만 대칭적인 연산들을 수행할 수 있기 때문인데, 프로토콜 설계자는 프로토콜을 강화하기 위해 연산들과 메시지들만 추가하면 된다. 반면, 평문 동등성을 피하기 위해서는 클라이언트/서버 양측에서 각각 비대칭적으로 동작하는 수학적 연산들을 매우 신중하게 선택해야 한다.

서버의 패스워드 데이터베이스를 공격자로부터 막는 것은 매우 좋은 방법이지만, 파일 허가나 접근 제어 리스트 같은 운영 체제 메커니즘을 벗어나지 못한다. 특정 지점에서 서버는 인증 수행을 위해 자신이 파일을 읽을 수 있어야 하므로, 완전한 패스워드 데이터베이스 내용에 접근하는 방법이 서버 어딘가에 존재해야 한다. 수십 년간 유닉스 사용자들은 평문-동등하지 않은 패스워드 메커니즘의 상대적 안전성에 의존해 왔으나, 최근의 운영 체제 벤더들은 평문 동등성의 위험을 인식하기 시작했다. 패스워드 파일을 누구나 읽을 수 있더라도, 공격자들에게 이러한 패스워드 파일이 전혀 유용하지 않도록 하는 시스템을 사용하는 것이 진정한 안전성의 시작일 것이다.

3. 전향적 안전성

패스워드-인증 프로토콜에 적용되는 전향적 안전성은 프로토콜의 다른 부분이 손상되더라도, 비밀 정보의 안전성에 영향을 미치지 않는 성질을 말한다. 이러한 전향적 안전성은 패스워드-인증 시스템이 인증 뿐 아니라 트래픽 암호화를 위한 세션 키 교환에 빈번히 사용되면서 그 중요성이 부각되고 있다. 아무리 안전한 패스워드-인증 시스템이라고 해도, 키나 패스워드가 손상될 가능성은 항상 존재하는데 사용자는 패스워드를 적어 놓거나 쉽게 추측 가능한

것을 선택할 가능성이 있고, 세션키들은 노출되거나 성능이 낮은 난수 발생기 또는 쉽게 추측 가능한 시드(seed)로 인하여 생성될 수 있다. (그림 4)는 키 교환을 수행하는 변형된 Challenge-Response 프로토콜이다.



(그림 4) 키 교환을 위한 변형된 Challenge-Response 프로토콜

클라이언트 A는 랜덤수 R을 생성하여 서버 B에게 전송하고, B 역시 랜덤수 S를 생성하여 A에게 보낸다. A와 B 양측은 자신들이 동일한 K를 갖고 있는지 확인하고, 확인이 되면 이 K를 세션키로 메시지 암호화 등에 사용할 수 있게 된다.

위의 그림에서 공격자가 P를 얻게 되었다고 가정하자. 공격자는 트래픽을 모니터링하여 R과 S 값을 얻고, h(R, S, P)를 계산함으로써 세션키 K를 획득할 수 있다. 이러한 공격 방법은 패스워드-체이닝(chaining) 현상 때문에 간과할 수 없다. A가 B의 시스템에서 패스워드를 변경하고자 할 경우, 먼저 로그인을 하고 정확한 명령어를 실행시키므로, 그 해당 세션을 복호화할 수 있는 사람은 A의 새로운 패스워드를 볼 수 있게 된다. 만일, 공격자가 A의 올드 패스워드를 알고 있다면, A가 패스워드를 변경할 때까지 올드 패스워드로 모든 암호화된 세션을 복호화할 수 있다. 계속해서 공격자는 새 패스워드로 패스워드가 변경될 때까지 세션들을 복호화할 수 있고, 결국엔 모든 A 세션 트래픽이 복호될 수 있는데 이러한 현상을 패스워드-체이닝이라 한다.

한편, 공격자가 세션키 K를 획득했다고 가정하자. 예를 들면, 미국의 암호 수출 제한으로 인하여 안전성이 충분하지 않은 40비트의 키 길이를 갖는 암호 소프트웨어가 사용된다면, 무작위적 공격으로 몇 시간 내에 깨질 것이다. 현재 사용되는 많은 시스템들은 안전한 연결을 먼저 구축하고 그 구축된 안전한 연결을 통하여 패스워드를 보내는 방식을 취하는데 이러한 경우, 세션키가 손상되면 패스워드는 바로 노출되게 된다. 공격자가 세션에서 K를 획득할 경우, P'을 추측하여 사전 공격을 수행할 수 있다. 즉, h(R, S, P')=? K 인지 비교한다. 이러한 사전 공격을 Denning-Sacco 공격이라 한다[2].

IV. 패스워드 인증 연구 동향

비대칭 및 공개키 암호 방식의 발전에도 불구하고, 강한 패스워드 인증은 암호학에서 매우 어려운 문제로 남겨져 있다. 현재 사용중인 대부분의 시스템은 약하고 쉽게 손상될 수 있는 패스워드 인증을 사용한다.

1. 약한 인증

일반적으로, "약한" 인증 시스템은 패스워드를 네트워크 상에 그대로 노출시키거나, 인증을 수행하는 동안 공격자가 패스워드를 유추하거나 추측할 수 있는 충분한 정보를 노출시키는 프로토콜을 사용한다. 패스워드를 평문 형태나 해싱된 형태로 전송하는 프로토콜이나 Challenge-Response 프로토콜이 이 경우에 해당한다. 유닉스 시스템에서 사용하는 해싱된 패스워드 전송은 검증을 위해 안전하게 패스워드를 서버에게 전송하는 방법이나, 패스워드 해싱 값 자체가 사용자로 위장하는 데 사용될 수 있으므로 안전성 측면에서 그리 바람직한 방법은 아니다. Challenge-Response 프로토콜의 경우, 서버가 사용자에게 전송하는 Challenge는 일종의 랜덤 스트링이고, 사용자가 서버에게 전송하는 Response는 Challenge와 패스워드에 기반한 함수이다. 공격자가 유효한 Challenge-Response 쌍을 얻는다고 해도, 그 다음 Challenge는 다를 것이고 그에 따른 Response도 다를 것이기 때문에 시스템에 접근하

는 데 도움을 주지 못하지만, 전 장에서 설명한 사전 공격, 평문 동등성 및 전향적 안전성 등 3가지 문제점에 대하여 안전하지 못한 단점이 있다.

2. 강한 인증

강한 암호화(encryption)가 지금까지 제안되었지만, 강한 상호 인증 프로토콜은 EKE[3]류의 알고리즘들이 나온 1990년대 이후에 제안되기 시작하였다.

가. EKE

EKE(Encrypted Key Exchange)는 패스워드 인증 수행을 위해 비대칭 암호 방식과 공개키 암호 방식을 결합한 것으로 사전 공격에 견디고 신뢰 기관이나 키-관리 없이 전향적 안전성을 제공하는 최초의 프로토콜이다. EKE는 공격자가 자신이 추측한 패스워드의 검증을 위한 정보를 충분히 얻지 못하도록 함으로써 사전 공격을 막는다. ElGamal[4] 공개키 암호 방식을 사용하는 일반적인 EKE 형태에서, 두 통신 당사자들은 분배된 패스워드를 대칭키 암호 방식의 키로 하여 키 재료를 암호화한다.

EKE의 가장 큰 단점은 평문 동등 메커니즘을 사용한다는 것으로 클라이언트나 서버가 동일한 비밀 패스워드나 해시값에 접근한다.

나. DH-EKE와 SPEKE

EKE는 다른 공개키 시스템과 함께 구현될 수 있으며, 각각은 사용된 알고리즘에 따라 특성을 갖게 된다. 가장 잘 알려져 있고 안전한 형태를 갖는 것이 Diffie-Hellman 키교환 프로토콜[5]과 유사한 DH-EKE(Diffie Hellman Exponential Key Exchange)이다. 두 프로토콜의 다른 점은 Diffie-Hellman 키 교환 프로토콜에서 교환되는 메시지가 분배된 패스워드로 암호화된다는 것이다.

SPEKE(Simple Password Exponential Key Exchange)[6]는 Diffie-Hellman 키 교환 방식에 기반하지만, 패스워드는 세션키 생성 함수에서 생성기(generator) 파라미터 선택에 사용된다. EKE와 SPEKE는 사전 공격에 안전하고 완전 전향적 안전성(Perfect Forward Secrecy: PFS)을 갖지만, 패스워드의 평문 동등성 문제는 해결하지 못하였다. 이러한 프로토콜들은 서버가 클라이언트와 동일한 패스워드를 알고 있다고 가정한다.

다. A-EKE

EKE를 설계한 Bellovin과 Merritt은 EKE 프로토콜의 강화된 버전인 A-EKE(Augmented EKE)를 제안하였다[7]. A-EKE 프로토콜에서는 서버가 사용자의 패스워드와 평문 동등하지 않은 정보를 저장한다. 이것은 사전 공격에 견디고 평문 동등성을 갖지 않는 유일한 프로토콜이지만, 평문 동등성을 피하려는 시도 때문에 전향적 안전성은 갖지 못한다.

라. SRP

SRP(Secure Remote Password) 프로토콜은 기존의 패스워드-기반 프로토콜들이 갖는 대부분의 문제들을 어느 정도 해결한 프로토콜이다[8]. SRP는 네트워크 상의 모든 수동적, 능동적 공격에 견디는 새로운 클래스의 강한 인증 프로토콜로서, 다른 기존의 키 교환 프로토콜과 인증 프로토콜들로부터 부분적 요소들을 도입하여 이를 변형하고 정제하였다. 이러한 결과로, SRP는 EKE 류 프로토콜들의 강도와 효율성을 유지하면서 그들이 가진 단점들을 극복하였다.

1) 표기 및 환경 설정

- N : 안전한 큰 소수(N=2q+ 1, q는 소수), 모든 연산은 모듈로 N상에서 이루어짐
- g : 모듈로 N상의 원시원소(generator)
- U : 사용자 이름
- p : Cleartext 패스워드
- H() : 일방향 해시 함수
- t : 안전성 파라미터
- u : 랜덤 스크램블링 파라미터

- s : 사용자의 salt
- a, b : 비밀 값(secret ephemeral values)
- A, B : 공개 값(public ephemeral values)
- x : 비밀키(p와 s로부터 유추됨)
- v : 패스워드 검증자(verifier)

여기서 salt는 사전 공격을 어렵게 하기 위한 일종의 랜덤 스트림으로, 일방향 함수로 계산하기 전에 패스워드와 concatenation한다. salt값과 일방향 함수 값이 함께 호스트의 데이터베이스에 저장되는데, 가능한 salt값이 매우 많을 경우, 공격자가 각각의 가능한 salt값 마다의 일방향 해시를 생성해야 하므로, 일반적인 패스워드에 대한 사전 공격을 효과적으로 막을 수가 있다.

2) 동작 과정

- 호스트는 다음과 같이 패스워드를 저장한다.
  $x = H(s, p)$ (s는 랜덤하게 선택됨)
  $v = g^x$
- 호스트는 패스워드 데이터베이스에 U, s, v를 저장한다.
- 사용자는 호스트에게 신분 증명을 위해 이름과 $A(=g^a)$를 전송한다(여기서 a는 랜덤수)(①).
- 호스트는 salt s와 $B(=v+g^b)$, 그리고 랜덤한 스크램블링 파라미터 u를 사용자에게 전송한다(여기서 b는 랜덤수)(②).
- 사용자는 $x(=H(s, p))$와 세션키 $S(=(B-g^x)^{(a+ux)})$, 그리고 $K(=H(S))$를 계산한다(③).
- 호스트는 세션키 $S(=(Av^u)^b)$와 $K(=H(S))$를 계산한다. 이로써 양측은 분배된 세션키 K를 갖게 된다. 양측은 인증을 위해 각기 자신들의 키가 매칭됨을 증명할 필요가 있다(④).
- 사용자는 M(=(H(H(N) XOR H(g), H(U), s, A, B, K))을 계산하여 호스트에게 전송한다(⑤).
- 호스트는 받은 M과 세션키를 이용하여 H(A, M, K)를 계산하여 사용자에게 전송한다(⑥).

(그림 5)는 위의 동작 과정을 나타낸 그림이다.

사용자(P)	g	호스트(U, s, v)
$A=g^a$ 계산		
	① U, A	
	→	$B=v+g^b$ 계산
	② s, B, u	
③ $x=H(s, p)$, $S=(B-g^x)^{(a+ux)}$	←	
K=H(S) 계산		④ $S=(Av^u)^b$, K=H(S) 계산
M= H(H(N) XOR H(g),	⑤ M	
H(U), s, A, B, K) 계산	→	
	⑥ H(A, M, K)	M 확인후, H(A, M, K) 계산
	←	

(그림 5) SRP 동작과정

3) SRP의 안전성 관련 장점

- 스누핑(snooping)에 안전하다. 패스워드 자체가 클리어한 상태나 또는 암호화된 형태로도 네트워크 상으로 전송되지 않는다.
- 재전송 공격(replay attack)에 안전하다. 인증시 교환된 어떠한 정보도 SRP를 사용하는 서버에 접근 허가를 얻기 위해 재 사용되지 않는다.
- 인증 과정 중에 세션키를 교환한다. 이 세션키는 사용자의 로그인 세션을 암호화하는 데 사용될 수 있으며, 이 세션을 스누핑과 악의적 능동 공격으로부터 보호한다.
- 상호 인증(mutual authentication)을 제공한다. 이것을 만족하기 위해서 양측은 그들의 비밀을 안전하게 지니고 있어야 한다.
- 교환된 메시지를 이용한 오프라인 사전 공격에 안전하다. 네트워크를 통해 교환되는 트래픽은 사용자 패스워드 추측을 검증하는 데 불충분하다.
- 완전 전향적 안전성을 제공한다. 공격자는 손상된 패스워드로 과거 세션을 복호화할 수 없다. 이것은 Denning-Sacco 공격에도 안전하다는 의미이다. 즉, 공격자는 손상된 패스워드로 사전 공격을 수행할 수 없게 된다.
- 호스트의 검증자 데이터베이스 손상에 견딘다. 물론 이러한 손상으로 인하여 시스템에 대한 사전 공격이나 호스트 위장 같은 몇몇 공격이 가

능하게 되지만, 패스워드 엔트리들은 사용자 패스워드 검증에만 사용되기 때문에 그다지 위협적이지는 않다(즉, 패스워드 엔트리들은 실제 패스워드와 평문 동등하지 않다). 이러한 패스워드 엔트리들은 공격자가 호스트에 직접 액세스하는 데 사용되지 않는다.

4) SRP의 기술적 및 실용적 장점

- 비교적 쉽고 구현하기 쉬운 멱승, 덧셈, 곱셈, 해싱 등으로 이루어지므로 매우 간단한 프로토콜이다.
- Diffie-Hellman 키 교환 프로토콜만큼 빠르고, 이를 통한 많은 최적화(예를 들면, 사용자가 느끼는 시간 지연을 줄이기 위한 병렬 연산 같은)가 SRP에 적용될 수 있다.
- 표준화, 구현 및 디버깅이 용이하다. SRP는 간단한 연산들의 집합이므로, 구성된 기능들을 수행하는 코드들을 찾기 쉽고, 그것을 어떤 클라이언트 어플리케이션과도 통합이 가능하다. 또한, SRP는 네트워크 트래픽과 관련하여 매우 경제적이므로, 표준화가 용이하다.
- 일반적으로 3개의 메시지를 사용한 전송이 강한 인증을 위한 최소라고 여겨진다. SRP는 (양측의 개인 식별 정보와 파라미터 메시지 전송은 제외) 4개의 메시지를 사용하나, 4번째 메시지는 상호 인증이 요구될 시에만 필요하다.

3. 비효율적인 인증

강하고 간편한 패스워드 인증 기술이 없었던 1980년대의 시스템 설계자들은 패스워드 안전성을 보증하기 위한 다른 기술들을 제안하였다. 대부분의 이러한 시스템들은 물론 완전한 패스워드 기반이 아니고, 원활한 운용을 위해 사용자측이나 관리자측, 또는 양측 모두에게 추가적인 오버헤드를 필요로 한다.

가. 일회용 패스워드

일회용 패스워드는 말 그대로 한 번만 사용되고 버려지는 패스워드를 말하며, 가장 잘 알려진 시스템으로 S/Key를 들 수 있다. 획득한 패스워드는 공격자에게는 무의미하므로, 안전성 관점에서의 장점은 명백하다. 비효율성 측면 역시 명백한데 사용자들은 패스워드 리스트나 진행중에 일회용 패스워드를 계산하는 클라이언트 소프트웨어를 유지, 관리해야 하고, 사용자가 패스워드 리스트를 모두 소모했을 때마다 이를 재생성해야 한다.

나. Kerberos

MIT 프로젝트 Athena에서 개발한 분산 인증 서비스로 인증은 중앙 인증 서버가 생성한 티켓을 통하여 승인된다. Kerberos는, 특정 지역이나 도메인에서 인증되는 모든 개체 당 하나씩인 비밀키 집합을 유지해야 하므로, 서버가 극도로 안전해야 한다. 이것은 전체 시스템과 그것에 연결된 모든 노드들과 사용자들에 대한 단일 포인트 실패를 의미한다. 또한 Kerberos는 많은 관리 오버헤드를 갖는 단점이 있다. 사용자와 호스트가 서로 다른 관리 도메인에 있을 경우 인증을 수행하지 못하므로, 모든 사용자에게 동일한 인증 서비스를 제공하는 중앙-관리 클러스터 네트워크에 적합하다. Kerberos 4 TGT 프로토콜에 대한 안전성이 제고되었고, Bellovin과 Merritt은 이 프로토콜이 사전 공격에 견디지 못함을 보였다. Kerberos 4는 비효율적이고 약한 안전성을 제공하나, Kerberos 5는 사전 공격의 위협을 제거할 수 있는, 사전 인증(pre-authentication) 방법을 도입하여 이 문제를 해결하였다.

V. 결 론

지금까지 패스워드-기반 인증 및 키 교환 프로토콜의 개념과 배경, 그리고 특성과 최근까지의 연구 동향에 대하여 살펴 보았다. 패스워드-기반 프로토콜은 사용자가 패스워드만을 기억하고 있으면 되므로 그 활용도가 매우 높고, 사용자 편의성 관점에서 장점을 가지므로 전자상거래의 많은 분야에 적용될 것으로 예상된다. EKE 류의 프로토콜 외에도 다수

의 패스워드-기반 프로토콜이 제안되었으며 지금까지도 연구가 활발히 진행되고 있다.

참 고 문 헌

[1] J.G. Steiner, B.C. Neuman, and J.I. Schiller, "Kerberos: An Authentication Service for Open Network Systems," *USENIX Conference Proceedings*, 1988, pp. 191 - 202.

[2] D.E. Denning and G.M. Sacco, "Timestamps in Key Distribution Protocols," *Communications of the ACM*, Vol. 24, No. 8, 1981, pp. 533 - 536.

[3] S.M. Bellovin and M. Merritt, "Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks," *Proceedings of the 1992 IEEE Computer Society Conference on Research in Security and Privacy*, 1992, pp. 72 - 84.

[4] T. ElGamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *Advances in Cryptology: Proceedings of CRYPTO 84*, Springer-Verlag, 1985, pp. 10 - 18.

[5] W. Diffie and M.E. Hellman, "New Directions in Cryptology," *Transactions on Information Theory*, Vol. IT-22, No. 6, 1976, pp. 644 - 654.

[6] D.P. Jablon, "Strong Password-only Authenticated Key Exchange," *ACM Computer Communications Review*, 1996.

[7] S.M. Bellovin and M. Merritt, "Augmented Encrypted Key Exchange: a Password-Based Protocol Secure Against Dictionary Attacks and Password File Compromise," TR, AT&T Bell Lab, 1994.

[8] T. Wu, "The Secure Remote Password Protocol," NDSS'98, *1998 Internet Society Symposium on Network and Distributed System Security*, 1998.