The Secure Remote Password Protocol

Thomas Wu
Computer Science Department
Stanford University
Tjw@cs.Stanford.EDU

November 11, 1997

Quote of the day:

“ Well begun is half done. “

Aristotle, Politics

Purpose:

Presenting a new password authentication and key-exchange protocol that is suitable for:
- authenticating users and
- exchanging keys
- over an untrusted network

Categories of authentication methods:

Something that:
- The user is (voiceprint identification, retinal scanners)
- The user has (ID cards, smartcards)
- The user knows (password, PINs)
  - Direct password authentication (used in this paper)

Direct Password Authentication

- Client: No persistent stored information
- User's password: memorized quantity. Only secret that available to client software
- Network(C-S): vulnerable to both eavesdropping and deliberate tampering by the enemy.
- No trusted third party (key server or arbitrator). Only original two parties.

AKE (Asymmetric Key Exchange):

- New family of authentication protocols
- Generalized form for a third class of verifier-based protocols.
- Uses swapped-secret instead of traditional shared-secret.
- Do not use symmetric encryption.
### Secure Remote Password (SRP):

- **Interpretation of AKE**
  - Simple, fast and highly secure
  - Uses Simplified MAC (Message Authentication Code) that is based on one-way hash functions. For verifying the session keys by two parties in a secure manner.
  - Resists dictionary attacks that mounted by passive or active network intruders.
  - Offers perfect forward secrecy (protects past session and passwords against the future compromises).
  - Stores user passwords stored in a form that not plaintext equivalent to the password (an attacker captures the password database cannot use it directly to compromise security and access to the host).
  - Combines the techniques of zero-knowledge proofs with asymmetric key exchange protocol.
  - Offers better performance than Augmented EKE (A-EKE, digital signatures) or B-SPEKE (secondary one-sided key exchange).

### Mathematical notation of SRP

- **Mathematical notation of SRP**
  - \( n \): A large prime number (All computations are performed modulo \( n \))
  - \( g \): A primitive root modulo \( n \) (generator in \( GF(n) \))
  - \( s \): A random string (user's salt)
  - \( P \): The user's password
  - \( x \): A private key (derived from \( P \) and \( s \))
  - \( v \): The host's password verifier
  - \( u \): Random scrambling parameter, publicly revealed
  - \( K \): Session key
  - \( H() \): One-way hash function
  - \( a, b \): Ephemeral private keys, generated randomly, not publicly revealed
  - \( A, B \): Corresponding public keys
  - \( m, n \): Two quantities (strings) \( m \) and \( n \) concatenated
  - \( GF(n) \): Finite field (All computations are performed in a finite field).

### Mathematical notation of SRP (cont.)

- **Mathematical notation of SRP (cont.)**
  - \( P(x) = g^x \): One way verifier-generating function.
  - \( Q(w, x) = w + ux \): Session key generation function.
  - \( S(w, x) = wx \): Session key generation function.
  - \( R(w, x) = uxu \): Mixing functions for private and public parameters.
  - \( Q(w, x) = w + ux \): Mixing functions for private and public parameters.
  - \( P(x) = g^x \): One way verifier-generating function.

### SRP Protocol

#### Carol

1. C sends S her username.
2. S looks up C's password entry and fetches her salt \( s \). He sends \( s \) to C. C computes his long-term key \( k \) using \( s \) and her real password \( P \).
3. C sends the random number \( x \) computes her ephemeral public key \( A \) and sends it to S \( A = g^x \).
4. C and S compute the common exponential value \( S \) if C's password \( P \) entered in step-2 matches the one she used to generate \( s \); then both values of \( S \) will match.
5. Both sides hash the exponential \( S \) into a cryptographically strong session key.
6. C sends S \( M_1 \) as evidence that he has the correct key. C also computes \( M_2 \) and verifies that it matches the one that C sent.
7. S sends C \( M_2 \) as evidence that he has the correct key. C also verifies \( M_2 \) and accepts only if it matches S's value.
8. S sends C \( M_2 \) as evidence that he has the correct key. C also verifies \( M_2 \) and accepts only if it matches S's value.

#### Steve

1. C sends S her username.
2. S looks up C's password entry and fetches her salt \( s \). He sends \( s \) to C. C computes his long-term key \( k \) using \( s \) and her real password \( P \).
3. C sends the random number \( x \) computes her ephemeral public key \( A \) and sends it to S \( A = g^x \).
4. C and S compute the common exponential value \( S \) if C's password \( P \) entered in step-2 matches the one she used to generate \( s \); then both values of \( S \) will match.
5. Both sides hash the exponential \( S \) into a cryptographically strong session key.
6. C sends S \( M_1 \) as evidence that he has the correct key. C also computes \( M_2 \) and verifies that it matches the one that C sent.
7. S sends C \( M_2 \) as evidence that he has the correct key. C also verifies \( M_2 \) and accepts only if it matches S's value.
8. S sends C \( M_2 \) as evidence that he has the correct key. C also verifies \( M_2 \) and accepts only if it matches S's value.

### Reduction to Diffie-Helman (DH)

- **Reduction to Diffie-Helman (DH)**
  - Mathematical notation of SRP protocol is similar to the DH problem.
  - The algorithm used to compromise the session key in SRP through a passive attack can be used to break a DH key exchange in polynomially-equivalent time.
  - This proof shows that SRP resists passive attack at least as well as DH protocol.
  - The algorithm can be:
    - Oracle \( Q \) that accepts the values \( A, B, u, g, n \), and \( x \) and computes session key \( S \).
    - \( S = g^{uxu} \)
    - \( Q(x, y, z, u, g, n) = g^{uxu} \)
    - It is difficult to compute \( g^x \) in \( GF(n) \) where \( g \) and \( x \) are given (as claimed in DH).
    - Therefore let \( u = g^x \), and \( x = (n-1)/2 \) DH oracle \( Q \) in terms of SRP oracle \( Q 
      - \( Q(A, B, n) = Q(A, B, g^{(n-1)/2}, 2, g, n, (n-1)/2) \) and \( A = g^x, B = g^y \) then
    - \( Q(x, y, z, u, g, n) = g^{uxu} \)
**Resistance to Denning-Sacco Attack**

- An intruder captures the session key from an evasdropped session and uses it to gain ability to access the user directly or conduct a brute-force search against the user's password.
- A-EKE requires the user to send a message which is dependent on both long term private key and the session key. This message enables the Denning-Sacco attack.

**Optimization of SRP**

- Less message rounds (instead of 3 round messages, 2 with Optimized SRP and 1.5 with One-Way Optimized SRP)
- Less execution speed (fastest verified-based protocol)

**Benefits of SRP:**

An attacker,

- with neither user’s password nor the host’s password file, cannot mount a dictionary attack on the password
- captures the host’s password file, cannot directly compromise user-to-host authentication and gain access to the host without an expensive dictionary search.
- compromises the host, does not obtain the password from legitimate authentication attempt.
- captures the session key, cannot use it to mount a dictionary attack on the password
- captures the user’s password, cannot use it to compromise the session keys of past sessions.