Secure Remote Password - A Secure Key Exchange Protocol

Kirti Bhandari
kirti_bhandari@cs.ucsb.edu
Department of Computer Science
University of California Santa Barbara

June 16, 2017

Abstract

Secure Remote Password (SRP) is an augmented password authenticated key agreement protocol based on secret key exchange. The protocol allows the participants to establish secure session keys without actual exchange of password. The paper discusses the SRP algorithm, cryptographic aspects leading to its security and also an analysis on security of the protocol.

1 Introduction

Password based authentication mechanisms should not rely on information stored on clients. Also, since the network is susceptible to eavesdropping, passwords should not be exchanged on the network. In addition, if we cannot rely on a third party key server, protocol design should be able to establish trust between client and server.

Secure Remote Password (SRP) protocol [5] is one such mechanism to provide password based authentication.

It also provides with a secure key exchange mechanism in order to establish secure sessions. It is an augmented Password Authenticated Key Exchange (PAKE) protocol, i.e. servers do not store any password-equivalent data. Hence any compromise of server data does not result into loss of security as the attackers cannot pose as clients.

In simpler terms, given two parties who both know a password, SRP (or any other PAKE protocol) is a way for one party to demonstrate to another party that they know the password, without sending the password or any other information from which the password can be derived.

2 Secure Remote Password

SRP is based on secret key exchange between client and the server.

The SRP protocol creates a large private key shared between the two parties in a manner similar to Diffie Hellman key exchange [3] based on the client side having the user password and the server side having a cryptographic verifier derived from the password. The shared public key is derived from two random numbers, one generated by the client, and the other generated by the server for every login attempt. For the requirement of encryption along with authentication, the SRP protocol is more secure than the alternative SSH protocol and faster than using Diffie Hellman key exchange with signed messages.

In this paper, we first discuss the details of the key establishment and exchange. Later we discuss cryptographic properties of the protocol. In the next section we discuss the features of the protocol which make it secure.

3 Protocol Design

The primary function of Asymmetric Key Exchange (AKE) is to exchange keys between two parties, the client and server, and to use this key to verify that both parties actually know their passwords. Unlike Encrypted Key Exchange, AKE does not encrypt any of the protocol flows. Instead, it uses predefined mathematical relationships to combine exchanged ephemeral values with established password parameters.

SRP follows AKE design to establish keys. Below are the parameters involved in SRP key calculations:

NOTE: | denotes concatenation.

- N - A large prime number. In SRP, computations are performed in Galois Field GF(N) [1]. q is a Sophie Germain prime [2], N = 2q + 1. N is called safe prime.
- g - A primitive root modulo N called a generator
- s - A random string used as the user's salt
- H() - One-way hash function
- I - username
- P - The user's password
- x - A private key derived from the password and salt. x can be generated in many different ways. For example, x = H(s | P) or x = H(s | H (I | ":" | P)) or x = H(s | I | P)

- v - The host's password verifier, computed as g^x
- u - Random scrambling parameter, publicly revealed
- a,b - Ephemeral (one time) private keys. These keys are generated randomly and not known publicly
- A,B - Public keys corresponding to a, b
- m,n - The two quantities (strings) m and n concatenated
- K - Session key
- k - H(N, g)

3.1 Algorithm:

Password establishment:
Initially, server computes x based on password for every user. Then it stores the values (I, s, v) in it's database or cache for every user. x is discard as it is password-equivalent text.

Key Computations: The cryptographically strong session key (K) calculations are done as follows:

1. $A = g^a$ where a is randomly generated. User $\rightarrow$ host - I, A
2. $B = kv + gb$ Host $\rightarrow$ User u, B
3. User, Host: u = H(A, B)
4. User: $S_{user} = (B - kg^x)^{(a+ux)} = (kv + g^b - kg^x)^{(a+ux)} = (kg^x - kg^x + g^b)^{(a+ux)} = (g^b)^{(a+ux)}$
5. User: $K_{user} = H(S_{user})$
6. Host: $S_{host} = (Av^u)^b = (g^a v^u)^b = [g^a (g^x)^u]^b = (g^{a+ux})^b = (g^b)^{(a+ux)}$
7. Host: $K_{host} = H(S_{host})$

Authentication: The actual authentication i.e. proof of password happens as follows:

1. User: $M_1 = H(A, B, K)$
2. User sends Host M_1 as a proof that she has the correct session key. Host computes M_1 himself and verifies it against the value User sent.
3. Host: $M_2 = H(a, M_1, K)$
4. Host sends User M_2 as evidence that he also has the correct session key. User also verifies M_2 herself, accepting only if it matches Host's value.

NOTES:

1. User will abort if she receives B = 0 (mod N) or u = 0.
2. Host will abort if he receives A (mod N) = 0.
3. User must show her proof of K (or "S") first. If Host detects that Host's proof is incorrect, he must abort without showing his own proof of K (or "S") i.e. M_2

4 Cryptographic Aspects

In the section, we discuss the cryptographic details which make SRP secure against attacks.

4.1 Cryptographic Security

1. SRP structure is similar to Diffie Helman problem. Since Diffie Helam problem is computations are infeasible for large values of parameters, intractability of DH can explain infeasibility of SRP compromise.
2. Since session key K is cryptographically strong, it prevents the attacker from guessing attacks against K.
3. N is a large safe prime. Security of SRP depends on the fact that private keys a and b are kept secret but A and B are publicly revealed. Computations in SRP are the form

   $$f = g^x$$

   where g is the generator and calculations are in GF(N). In order to recover x from any f, one needs to take discrete logarithm for large values of N.

   Finding discrete log is a computationally challenging problem. This leads to security of private keys like a and b and hence makes the protocol secure.

   Hence, In order to maximize the security, it is suggested to use large prime values of N
4. Additionally, using a non-smooth prime N leads to even difficult discrete log computations. N is considered non-smooth prime, if factors of (N-1) are not all small.

   In general, probability of generating a non-smooth prime is high, and hence large prime values for N can be easily generated.

   It is recommended to use

   $$N = 2q + 1$$

   where q is a prime. N of this form is called a safe prime and resists attacks like subgroup confinement attack [4]

5. B is computed as $g^x + g^b$ rather than g^b. This is because, using g^b as B leaves the protocol vulnerable to attacks.

   For example, consider a scenario where attacker G is trying to post as a host and captures salt s from a valid session (since s is public).

   The user sends her username I to attacker G. Attacker G sends the salt s it has snooped. User sends her public exponent A to attacker G. Attacker can randomly pick b and u and computer own public exponent B = g^b User then computes session key $S = B^(a + ux)$, computes session key K = H(S) and sends it as a proof to attacker G.

   Attacker G now has A and b, along with a proof of K from user. She can guess at a password p', compute x' from it and then v' from that, construct S' as $S' = (Av'^u)^b$, and finally K' = H(S'), and check it against user's proof of the real K. If they match, the guessed password is correct.

   Hence the protocol becomes vulnerable to dictionary attacks where attacker can try various passwords to guess one.

   This attack is possible because attacker who doesn't know $v = g^x$ can present B, it is desirable to add a component for x in B.

4.2 Constraints on Parameters

Following checks must be performed for security of SRP

1. N is a large safe prime. As discussed in previous subsection, N should be large and computed as 2q+1 where q is prime
2. g is primitive root of GF(N).
3. $A > 0$ This prevents the server's session key from being set to 0.
4. $B > 0$
5. $a, b > log[g]n$

5 Correctness

If correct password is supplied, both parties will always agree on a session key as shown in Key Computations step 4 and step 6.

6 Security Analysis

For security, an attacker trying to pose as a host should not be able to gain access to the host by observing messages exchanged during successful session establishment. Below are the key features which make SRP secure:

1. Since session key K is cryptographically strong, it prevents the attacker from guessing attacks against K.
2. Information about password P or corresponding key x is not revealed during successful. This prevents the attacker from guessing passwords based on exchanged messages.
3. An attacker with neither the user's password nor the host's password file cannot conduct a dictionary attack on the password. Mutual authentication is achieved in this scenario.
4. An attacker who captures the host's password file cannot directly compromise user-to-host authentication and gain access to the host without an expensive dictionary search.
5. An attacker who captures the session key cannot use it for dictionary attacks on the password.
6. Since passwords are not stored on host, an attacker who compromises host cannot get access to passwords from previous authentication attempts.
7. An attacker with access to the user's password cannot use it to compromise the session keys of past sessions as the session keys are based on multiple randomly generated components which cannot be reproduced.

7 Conclusions

We discussed Secure Remote Password based key exchange and authentication protocol. We have discussed the algorithm and the cryptographic properties which make the computations secure. Lastly we have conducted a security analysis on the protocol and identified the aspects of the protocol which lead to its security.

References

[1] Gf. `https://en.wikipedia.org/wiki/Finite_field_arithmetic`.

[2] Sophie germain. `https://en.wikipedia.org/wiki/Sophie_Germain_prime`.

[3] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[4] David P. Jablon. Strong password-only authenticated key exchange. *SIGCOMM Comput. Commun. Rev.*, 26(5):5–26, October 1996.

[5] Thomas D Wu et al. The secure remote password protocol. In *NDSS*, volume 98, pages 97–111, 1998.