STRONG PASSWORD PROTOCOLS

Lamport's Hash, EKE, SPEKE, PDM

LAMPORT'S HASH (I)

- Server B authenticates user A
- A has only a password and A's workstation has no private key
- Secure against eavesdropping and attacks on B's database

LAMPORT'S HASH (II)

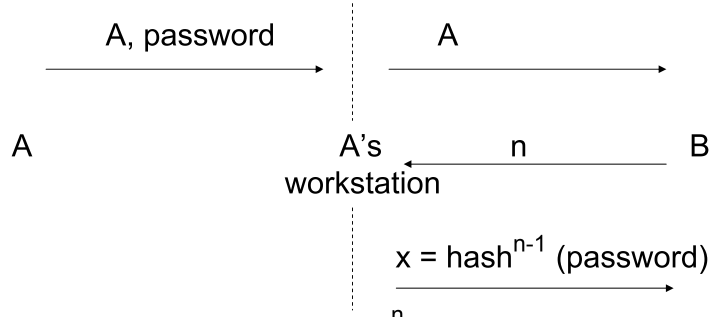
- A chooses a password and a number n (≈ 1000)
- A computes hash^n (password)
- For each user B stores

  - username (transmitted by A)

  - integer n decremented after each authentication

  - hash^n (password)

LAMPORT'S HASH (III)



Initially B knows n and hash^n (password)
On receipt of x, it hashes x and compares result to hash (password)
If equal, B replaces hash^n (password), n with x, n-1

LAMPORT’S HASH (IV)

- May enhance with “salt”
- A chooses “salt” and sends to B together with n and hash^n(password salt)
- B sends (n, salt) to A on request
- Allows A to use same password on multiple servers
- Allows A to retain same password when n = 1

LAMPORT’s HASH - ISSUES

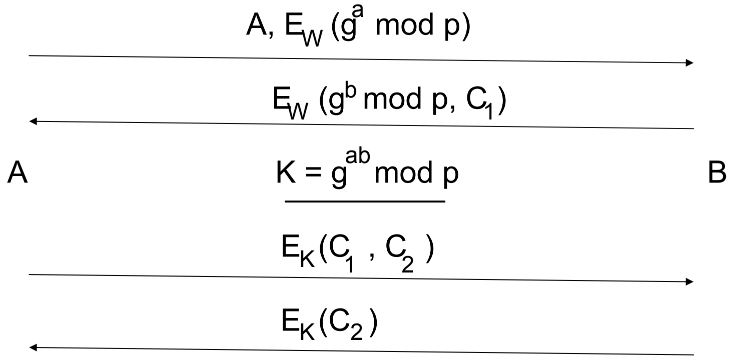
- Unless “salt” is used reinstall password when n = 1
- B is not authenticated by A
- Vulnerable to “small n attack”
- May employ without workstation software (i.e. user is provided with table of hash^n (password) and n)

Encrypted Key Exchange (EKE)

- Uses weak secret derived from password
- Protects against dictionary attack
- Multiple incorrect guesses should raise alarm

EKE (II)

share weak secret W = f (password)



A, E_W (g^a mod p)

E_W (g^b mod p, C_1)

A

$\underline{K = g^{ab} \text{ mod } p}$

B

$E_K(C_1 , C_2)$

$E_K(C_2)$

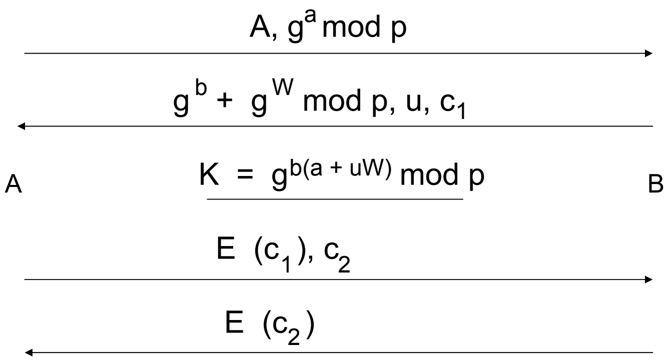
Simple Password Exponential Key Exchange (SPEKE)

- Uses weak secret W in place of g in Diffie Hellman exchange in EKE
- Exchanges W^a mod p and W^b mod p
- Agreed key is K = W^{ab} mod p

Password Derived Moduli (PDM)

- Modulus p = f (password)
- g = 2
- Agreed key K = 2^{ab} mod p

Secure Remote Password (SRP)



A, g^a mod p

$g^b + g^W$ mod p, u, c_1

A

K = $g^{b(a + uW)}$ mod p

B

E (c_1), c_2

E (c_2)

SRP Details

- A computes W from password
- B stores g^W mod p and associates with A
- g and p are fixed for the system
- a and b are chosen by A, B respectively
- challenges c_1 and c_2 are chosen by A, B respectively
- B additionally chooses a 32-bit number u
- SRP is documented in RFC 2945 and is common in IETF protocols