Trust Models—Public Key vs. Symmetric Key

Public-Key World

- Hierarchy of Trust.
- Cross Domain Certification.
- Web of Trust.

Authentication

Alice $\longrightarrow$ Bob.

Concerns: Eavesdropping, Exposing secrets on server.
Goal: No secrets on the server + foil eavesdropping.
Methods: Passwords, One-Time Passwords, Challenge-Response protocols, Zero-Knowledge authentication.

In typical environments, authentication has to be combined with session key exchange protocol (for encryption/authentication) to foil session hijacking.

Passwords

- Used to authenticate people.
- Have low entropy (≈ 25 bits).
- Susceptible to eavesdropping, replay to server.

A $\longrightarrow$ B

$$\boxed{\begin{array}{c} pwd_1 \\ pwd_2 \\ \vdots \\ pwd_n \end{array}}$$

- Never store passwords on the server.
- Store the password hash instead. Don't need the ability to invert. If this file is exposed an adversary can mount a dictionary attack.

  for(every word w in dictionary) { compute $h(w)$ }
  look for $h(w)$ in the password hash file.

Unix uses a modified DES algorithm with 12 bits of salt–a two-char string from the set `[a-zA-Z0-9./]`. It is used to perturb the algorithm in one of 4096 different ways. Usually encrypts 0 with the key 25 times. The value stored in the password file is a series of 13 printable ASCII characters (the first two characters are the 12-bit salt itself and the remaining 11 characters encode the 64-bit encryption of 0). See crypt(3).

Suppose 10M words in dictionary, 12 bit salt: then we have $10M \times 4K = 40G$ encrypted passwords. Assuming average length of 8 bytes gives $320GB$.

If one encryption is done in one μs, the dictionary can be encrypted in $40 \times 10^9 \times 10^{-6} = 4 \times 10^4$s ≈ 10 hours.

Passwords—Salting

Salt makes dictionary attack harder. An attacker must hash every word in the dictionary 2^{12} times. Logically, it looks like:

Alice	salt_a	$h(P_a \| \| \text{salt}_a)$
Bob	salt_b	$h(P_b \| \| \text{salt}_b)$

Secret Salt:

Store Alice$|\text{salt}_a|h(P_a||\text{salt}_a||\text{salt}_{a'})$

$\text{salt}_{a'}$ is 4 bits.To verify a user's password, the system tries all 2^4 combinations of $\text{salt}_{a'}$. Attacker's work goes up by 16.

Biometrics

One-Time Passwords

Lamport hash (S/Key) based on hash-chain [KPS02].

- Alice remembers a password p.
- Bob (server) remembers $(user, n, h^n(p))$.
- $A \rightarrow B$: A (I am Alice)
- $B \rightarrow A$: n
- $A \rightarrow B$: $x = h^{n-1}(p)$

Pros and Cons:

- Can only log in a finite # of times.
- No mutual authentication.
- Small n attack. What if Alice can be tricked into revealing $h^{50}(p)$.

References

[KPS02] Charlie Kaufman, Radia Perlman, and Mike Speciner. *Network Security—Private Communication in a Public World*. Prentice Hall PTR, 2nd edition, 2002.