COMP 3704 Computer Security

Christian Grothoff
christian@grothoff.org

http://grothoff.org/christian/

UNIVERSITY OF DENVER

Motivation

- Finding efficient protocols for problems can be a hard problem
- Security relies on good protocols, validation can be difficult
- Protocols can address surprisingly hard problems
- Protocols are fun to study

Interlock Protocol

1. Alice and Bob exchange their public keys
2. Alice sends Bob half of $E_{B_{pub}}(M_1)$
3. Bob sends Alice half of $E_{A_{pub}}(M_2)$
4. Alice sends Bob the other half of $E_{B_{pub}}(M_1)$
5. Bob sends Alice the other half of $E_{A_{pub}}(M_2)$
6. Both decrypt each other's messages with their private keys

Hash-based Authentication

1. Alice sends host her password P
2. Host compares $H(P)$ with database of hashed passwords

Salt

- Need to prevent Mallory from building database of all (common) passwords (**dictionary attack**)
- **Salt** is a random string that is concatenated with the password before hashing.
- Database contains salt S and hash $H(P + S)$

$\Rightarrow$ Mallory needs larger database

UNIVERSITY OF DENVER

Neuman-Stubblebine

1. Alice sends A, R_A to Bob.
2. Bob sends $B, R_B, E_B(A, R_A, T_B)$ to Trent, where T_B is a timestamp and E_B uses a key Bob shares with Trent.
3. Trent generates random session key K and sends $E_A(B, R_A, K, T_B), E_A(A, K, T_B), R_B$ to Alice where E_A uses a key Alice shares with Trent.
4. Alice decrypts and confirms that R_A is her random value. She then sends to Bob $E_B(A, K, T_B), E_K(R_B)$.
5. Bob extracts K and confirms that T_B and R_B have the same value as in step 2.

Denning-Sacco

1. Alice sends A, B to Trent
2. Trent sends Alice $S_T(B, K_B), S_T(A, K_A)$
3. Alice sends Bob $E_B(S_A(K, T_A)), S_T(B, K_B), S_T(A, K_A)$
4. Bob decrypts, checks signatures and timestamps

Lessons Learned

- Do not try to be too clever, over-optimization is often the cause for vulnerabilities
- Which optimizations you can do (and which optimization actually matter) depends on your assumptions (adversary model, system capabilities)
- Which protocol to use depends on your performance goals and communications capabilities (all-to-all communication, trusted party, latency, bandwidth and computational constraints)

Secret Splitting

1. Trend generates a random key K of the same size as the secret M and computes $M xor K = E$

2. Trend gives Alice K.

3. Trend gives Bob E.

Secret Sharing

- Goal: distribute secret S among n people, so that $k < n$ people can restore S
- Shamir: use polynominal of degree $k - 1$ (over finite field modulo p)
- Blakley: use point in m-dimensional space, share $m-1$-dimensional hyperplanes

Timestamping

1. Alice transmits $H(M)$ to Trent.

2. Trend sends $S_T(H(M), T_T)$ to Alice (where T_T is the time Trent received the message).

Questions

?

Problem

Alice wants to run a time-stamping service. Trent has a highly accurate atomic clock, but cannot be involved for each customer due to his high cost.

Alice is willing to buy hardware with a clock for her startup, but she cannot afford an accurate atomic model. How can Alice still enter the time-stamping business?

Problem

After her initial success in time-stamping, Alice has expanded her company and hired Bob, Carol and Dave. A customer asks the company to guard a secret; however, he is unwilling to trust any single person to be able to obtain the secret on their own. However, if Alice and one of her employees get together, they are supposed to be able to obtain the secret. Also, in the case that Alice is kidnapped, the other employees should together be able to obtain the secret, instead of being forced to pay Alice's ransom.

Problem

Describe possible attacks on this protocol:

1. Alice transmits $A, S_A(E_{B_{pub}}(K, R_A))$ to Bob.
2. Bob transmits $B, E_K(R_A)$ to Alice.
3. Their secure, authenticated exchange is then:
   (a) Alice sends $E_K(i_A, M_A^{i_A}, H(i_a, M_A^{i_A}))$ to Bob.
   (b) Bob sends $E_K(i_B, M_B^{i_B}, H(i_B, M_B^{i_B}))$ to Alice.

A real-world Protocol

1. Alice sends $A, S_A(T_A, E_{B_{pub}}(K_A, H(K_A), R_A))$ to Bob.
2. Bob verifies that T_A is larger than the last T_A and sends $B, S_B(T_B, E_{A_{pub}}(K_B, H(K_B), R_A, R_B))$ to Alice.
3. Alice verifies monotonicity of T_B and sends $E_{K_A}(R_B)$ to Bob. Their secure, authenticated exchange is then:
   (a) Alice sends $E_{K_A}(i_A, M_A^{i_A}, H(i_A, M_A^{i_A}))$ to Bob.
   (b) Bob sends $E_{K_B}(i_B, M_B^{i_B}, H(i_B, M_B^{i_B}))$ to Alice.

UNIVERSITY OF DENVER