Public Key Infrastructure (PKI) and Pretty Good Privacy (PGP)

EJ Jung
\[ \langle \text{Py}, \text{yahoo.com} \rangle : \text{Verification} \]

\[ \text{Website} \rightarrow \text{Yahoo.com} \]

\[ \text{Challenge} \rightarrow \text{RIP} \langle C \rangle = RC \]

\[ \text{fake website} \rightarrow \text{your browser} \]

\[ \text{Proof} \langle RC \rangle \rightarrow C? \]
Application

Transport: src - dst
Network (Internet) - routing
Access / Medium / link layer
physical

your browser → yahoo.com
Advantages of Public-Key Crypto

- Confidentiality without shared secrets
  - Very useful in open environments
  - No “chicken-and-egg” key establishment problem
    - With symmetric crypto, two parties must share a secret before they can exchange secret messages
- Authentication without shared secrets
  - Use digital signatures to prove the origin of messages
- Reduce protection of information to protection of authenticity of public keys
  - No need to keep public keys secret, but must be sure that Alice’s public key is really her true public key
Disadvantages of Public-Key Crypto

- Calculations are 2-3 orders of magnitude slower
  - Modular exponentiation is an expensive computation
  - Typical usage: use public-key cryptography to establish a shared secret, then switch to symmetric crypto
    - We’ll see this in IPSec and SSL
- Keys are longer
  - 1024 bits (RSA) rather than 128 bits (AES)
- Relies on unproven number-theoretic assumptions
  - What if factoring is easy?
    - Factoring is believed to be neither P, nor NP-complete
Encryption using Public-Key system
Authentication using Public-Key System
Problem: How does Alice know that the public key she received is really Bob’s public key?
Distribution of Public Keys

- Public announcement or public directory
  - Risks: forgery and tampering

- Public-key certificate
  - Signed statement specifying the key and identity
    - \( \text{sig}_{\text{Alice}}(\text{“Bob”}, \text{PK}_B) \)

- Common approach: certificate authority (CA)
  - Single agency responsible for certifying public keys
  - After generating a private/public key pair, user proves his identity and knowledge of the private key to obtain CA’s certificate for the public key (offline)
  - Every computer is pre-configured with CA’s public key
Using Public-Key Certificates

Authenticity of public keys is reduced to authenticity of one key (CA’s public key).
Typical Digital Signature Approach
Hierarchical Approach

- Single CA certifying every public key is impractical
- Instead, use a trusted root authority
  - For example, Verisign
  - Everybody must know the public key for verifying root authority’s signatures
- Root authority signs certificates for lower-level authorities, lower-level authorities sign certificates for individual networks, and so on
  - Instead of a single certificate, use a certificate chain
    - \( \text{sig}_{\text{Verisign}}("UI", PK_{UI}), \text{sig}_{UI}("EJ Jung", PK_E) \)
  - What happens if root authority is ever compromised?
Revocation of Certificates

Reasons for revocation:

- The user's secret key is assumed to be compromised.
- The user is no longer certified by this CA.
- The CA’s certificate is assumed to be compromised.
X.509 CA Hierarchy
Alternative: “Web of Trust”

- Used in PGP (Pretty Good Privacy)
- Instead of a single root certificate authority, each person has a set of keys they “trust”
  - If public-key certificate is signed by one of the “trusted” keys, the public key contained in it will be deemed valid
- Trust can be transitive
  - Can use certified keys for further certification

Alice → Friend of Alice → Friend of friend → Bob

\[ \text{sig}_{\text{Alice}}(\text{"Friend"}, \text{Friend’s key}) \]

\[ \text{sig}_{\text{Friend}}(\text{"Foaf"}, \text{Foaf’s key}) \]

I trust Alice