Summary From the Last Lecture

- Authorization via symmetric crypto
- Key exchange
  - Using asymmetric crypto
  - Using symmetric crypto with KDC
    - KDC shares a key with every participant
    - Authenticates participants, generates session (shared) keys for them to talk to each other
    - Gives the requester a "ticket" – session key, requester's ID and possibly expiration time, encrypted by the key that the server shares with KDC
    - Ticket and session key encrypted by the key that the requester shares with KDC
- Needham–Schroeder, Kerberos

Needham–Schroeder Key Exchange

Third-party authentication service
- Distributes session keys for authentication, confidentiality, and integrity
- Problem: replay attack in step 3
  - Fix: use timestamps

Kerberos

- Introduce Ticket Granting Server (TGS)
  - Issues timed keys to resources
- Users log on to authentication server (AS)
- AS + TGS = KDC
- Uses timestamps with a lifetime instead of nonces
  - Fixes freshness problem from Needham–Schroeder

Public Key Distribution

- Public key is public but ...
  - How does either side know who and what the key is for?
  - Does this solve key distribution problem?
    - No – while confidentiality is not required, integrity is
- Still need trusted third party
  - Digital certificates – certificate authority (CA) signs identity+public key tuple with its private key
  - Problem is finding a CA that both client and server trust

Digital Certificates

- Everyone has Trent’s public key
- Trent signs both Alice’s and Bob’s public keys – he generates public–key certificate
- When they receive keys, verify the signature
- Mallory cannot impersonate Alice or Bob because her key is signed as Mallory’s
- Certificate usually contains more than the public key
  - Name, network address, organization
- Trent is known as Certificate Authority (CA)
Certificate-Based Authentication

- Authentication steps
  - Verifier provides nonce, or a timestamp is used instead.
  - Principal selects session key and sends it to verifier with nonce, encrypted with principal's private key and verifier's public key, sends principal's certificate too.
  - Verifier validates certificate
  - Verifier checks signature on nonce

Key Exchange Examples

- PGP (Pretty Good Privacy)
  - "Web of Trust"
    - "As time goes on, you will accumulate keys from other people that you may want to designate as trusted introducers. Everyone else will each choose their own trusted introducers. And everyone will gradually accumulate and distribute with their key a collection of certifying signatures from other people, with the expectation that anyone receiving it will trust at least one or two of the signatures. This will cause the emergence of a decentralized fault-tolerant web of confidence for all public keys."

Key Exchange Examples

- X.509 standard
  - Hierarchical model
  - A handful of trusted CAs can issue certificates to others

Key Exchange Examples

- SSH
  - User keys exchanged out of band
  - Weak assurance of server keys
    - Is this the same host you spoke with last time?

Recovery From Exposed Keys

- Revocation lists (CRL’s)
  - Long lists
  - Hard to propagate
- Lifetime / Expiration
  - Short life allows assurance of validity at time of issue but increases cost of key distribution
- Real-time validation
  - Online Certificate Status Protocol (OCSP)
  - Single source of the compromised key list
  - Clients check suspicious keys and hash replies

Group Key Management

- Group key vs. Individual key
  - Proves that one belongs to the group vs. proving an individual identity
  - E.g., used for multicast messages
Group Key Management

- Revoking access
  - Change keys, redistribute
- Joining and leaving groups
  - Does one see old messages on join or is the key changed - backward secrecy
  - How to revoke access - forward secrecy
- Robustness
  - Coping with network partitioning
- Efficiency
  - Cost of use, verification, exchange

Group Key Management

- Centralized
  - Single entity issues keys
  - Optimization to reduce traffic for large groups
  - May utilize application specific knowledge
- Decentralized
  - Employs sub managers
- Distributed
  - Members do key generation
  - May involve group contributions