Computer and Network Security: Asymmetric Key Distribution

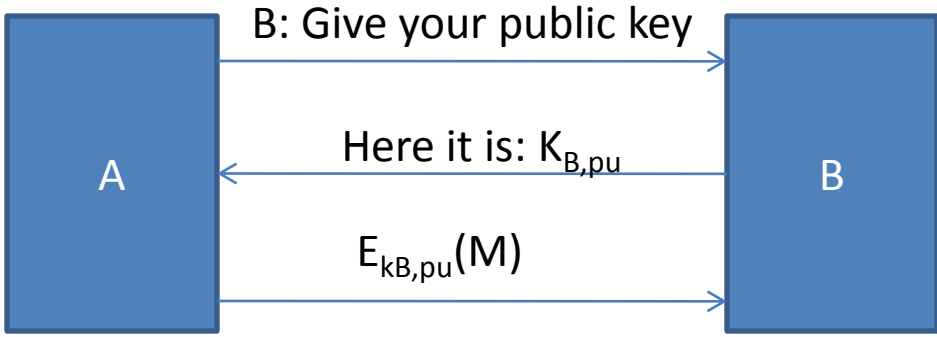
Kameswari Chebrolu

Outline

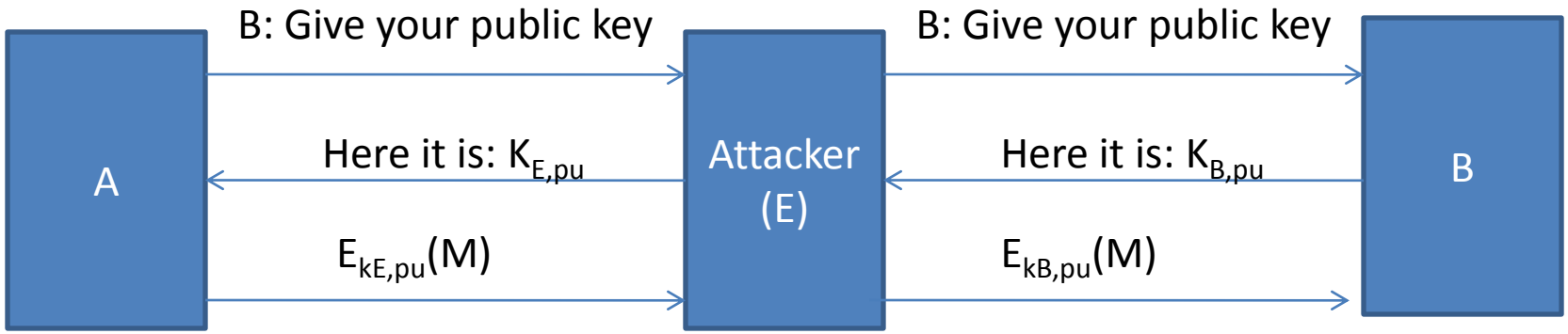
- Long-term Key Management
  - Symmetric and **Public key systems**
- Authentication Protocol
  - Password based
  - Short-term/session key establishment
  - Confidentiality/Integrity of data
  - One way, two-way and mediated authentication

Background

- Does this work?
  - NO
- Susceptible to Man-in-the-middle attack
  - Even B can be unaware of attack
  - Attack works for two-way conversation also



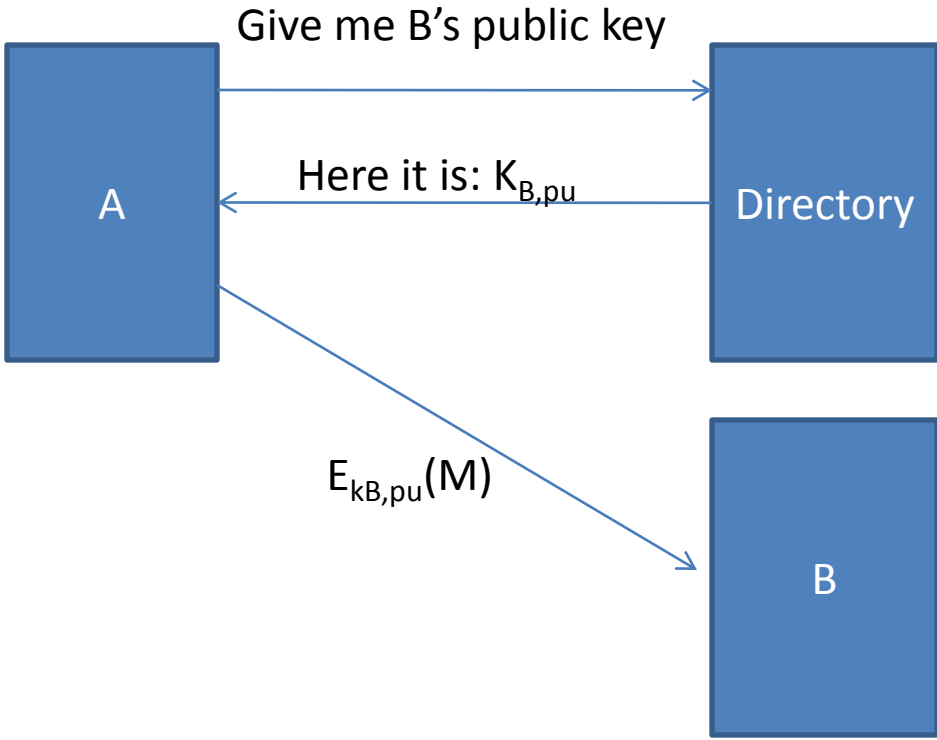
MITM Attack



Challenge: How does A authenticate B's public key?

Version-1: Trusted Directory

- A trusts Directory to give correct information
  - Directory maintains name to public key mapping
- Still susceptible to MIMT attack



Version-2: Trusted Directory

- A trusts directory
- A has directory's public key
  - To communicate securely with Directory to get B's public key
  - Directory's public key hardcoded into apps that use the directory

Drawbacks

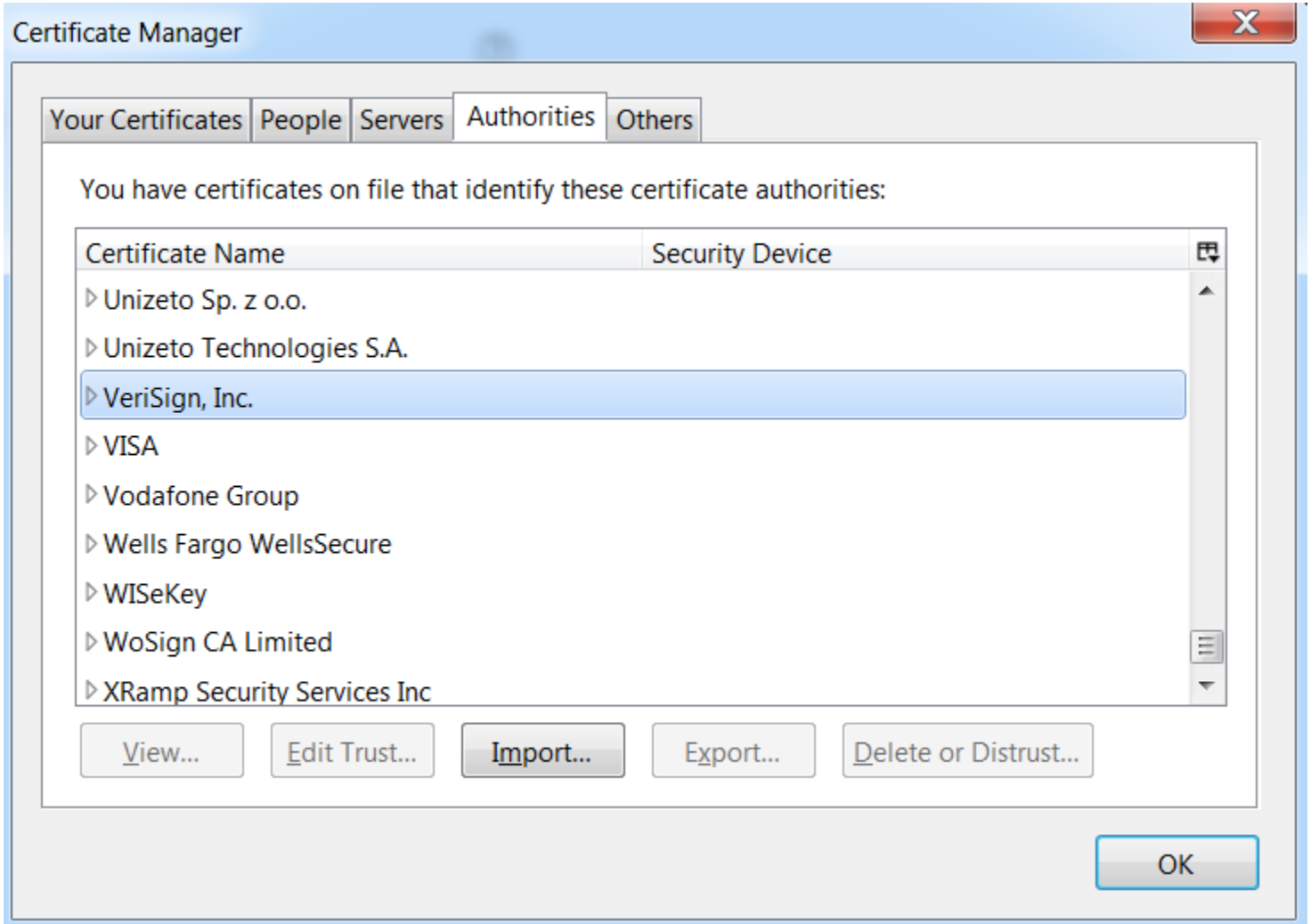
- Absolute trust in the directory
- Directory can become bottleneck
  - Has to handle many requests efficiently
- Directory can be single point of failure
  - Target of DoS attacks
- Directory has to be online

Version3: Digital Certificates

- A trusted entity signs [public key, key owner identifier, expiration time, serial number] using its private key → digital certificate
  - No need to contact directory service
  - Certificate can be got by any means
    - Google it, Get from B itself
- Trust still essential (as also the trusted entity's public key)
- Last 3 drawbacks are overcome
  - Scalability, single point of failure and being online

Public Key Infrastructure (PKI)

- Certificate Authority (CA) issues such digital certificates; very widely used
  - E.g. Verisign
  - Web browsers configured with list of many trusted Cas
    - E.g. Firefox → Options → Advanced → Certificates → View Certificates


Certificate Manager
Your Certificates
People
Servers
Authorities
Others
You have certificates on file that identify these certificate authorities:
Certificate Name
Security Device
Unizeto Sp. z o.o.
Unizeto Technologies S.A.
VeriSign, Inc.
VISA
Vodafone Group
Wells Fargo WellsSecure
WISeKey
WoSign CA Limited
XRamp Security Services Inc
View...
Edit Trust...
Import...
Export...
Delete or Distrust...
OK

Example

- www.iitb.ac.in uses http, say IITB wants to move to https (access via TLS)
- IITB needs to buy a certificate from a CA
  - Links www.iitb.ac.in (domain) with a public key
- When a user types https://www.iitb.ac.in
  - Browser asks IITB's website for a copy of its certificate
  - Browser verifies certificate using public key of CA who issued the certificate
  - Checks domain in certificate matches url typed in browser
  - Establishes secure connection to IITB web server using public key in the certificate

Multiple CAs

- A's CA is different from B's CA and A knows only its CA's public key
- How can A verify B's public key?
- The two CAs can issue certificates to each other
  - A gets a certificate, signed by its CA, specifying public key of B's CA (say P1)
  - A gets B's certificate specifying public key of B as P2 signed by P1
  - Now A can verify B's public key
- One can form chain of certificates of any length (much like with KDCs)

Key Revocation

- Simpler in KDC; just delete the key
- More complex with CAs
- Certificate validity: How long an expiration time?
  - Too short → too much overhead
  - Too long → More damage
- Certificate Revocation List (CRL): Lists serial number of certificate that are revoked
  - CA dates and signs the list
  - Users download latest copy periodically and check
  - CRL servers susceptible to DOS attacks

Other Approaches

- Web of Trust: Proposed by PGP for email encryption; not very popular
  - Any one can issue certificates to any other
  - Find multiple paths between sender and receiver
  - More paths/shorter paths → more trust
- Leap of faith/TOFU: Used by SSH; based on usable security
  - Take a leap of faith and accept public key received for first time
  - Second time on, use the same public key and warn if key changed

Summary

- Long-term key management feasible via
  - KDCs for symmetric keys
  - PKI/CAs for asymmetric keys (more popular)
  - Other approaches: web of trust, TOFU
- Both support multiple ‘trusted entities’ and key revocation mechanism
- Next: Authentication protocols
  - Assume long-term keys in place
  - Within: Short-term session key establishment, confidentiality and integrity