Key Management and PKI

Introduction to IT Security

Otago Polytechnic
Dunedin, New Zealand

From last week: public key cryptography

- Recall that we have two kinds of keys:
  1. Public keys
  2. Private keys
- Data encrypted with one key are decrypted with the other.
- If I encyrpt a message with my private key, you **know** it came from me.
- If you encrypt a message with my public key, you **know** that only I can read it.

How can we *know* these things?

- Last week I compromised someone's public key during our lab.
- Public key crypto depends on your ability to **trust** my public key.
- We need some mechanism to build that trust.

DIY approach: key signing parties

- We can all meet, face-to-face, confirm our identities, and exchange public keys.
- You go home afterward and *sign* the keys you received - meaning you verify (to yourself) their authenticity.

Web of trust

- Suppose that you and I exchange keys, so now you trust my key.
- Suppose that I also have and trust Darrell's key, but you do not.
- I can sign Darrell's key with my own and pass it on to you.
- Since you trust my key, and I've told you that I trust Darrell's key, you can now trust Darrell's key.

How do we scale this up?

This is great, but it won't scale. Amazon.com can't rely on everybody getting a personal introduction.

Certificate Authorities (CAs)

- A certificate authority is an organisation that acts as a *trusted third party* that issues cryptographically signed *digital certificates*.
- Examples include Symantec, Comodo SSL, and GoDaddy.
- If you need a cetrtificate, for example to run a secure web site, you verify yourself to a CA, pay a fee, and get a certificate.
- CA's also maintain *Certificate Revocation Lists* - lists of certificates that are no longer valid.

X.509

Digital certificates generally follow the X.509 standard.
Example:
`http://en.wikipedia.org/wiki/X.509#Sample_X.509_certificates`

Questuions?

We will see some client details of digital certificates and PKI in the lab.