CE0973a - Issues in Network Security
1: Introduction, Public Key Crypto

James A Sutherland

Abertay University

Monday, 11th January 2016

Course Overview

Two equally important components

- Exam
- Coursework (portfolio of weekly tasks)
  - Lecture: Mondays, 11-12, room 3032:
    James Sutherland, j.sutherland@abertay.ac.uk
  - Lab: Tuesdays, 9-11, 4510 (Netlab)
    Mark Bremner, 1100456@live.abertay.ac.uk
  - Feedback (week 7): lab session only, discuss coursework

Security

- Physical
- Personal
- Operational
- Communications
- Network

CIA

Some ISO standards for security best practices, management[1]

- Confidentiality: only the right people can read it
- Integrity: only the right people can write it
- Availability: they can access it when they need to

[1] http://iso-17799.safemode.org/

Five Pillars

- Authentication: prove you are Fred
- Authorisation: what is Fred allowed to do?
- Privacy
- Integrity
- Non-repudiation

Encryption and Security

- Hashing
- Symmetric (secret key)
- Asymmetric (public key)

Hashing, Message Digests

Turn any size block of data into a fixed size number. Easy to calculate, very hard to find a message which has a given hash value.

- MD5 (Merkle–Damgård, 128 bits)
- SHA-1 (Secure Hash Algorithm, 160 bits)
- SHA-2 (224/256/384/512 bits)

Hashing Applications

- Password storage
- File integrity checking
- Signatures and certificates

Secret Key Encryption

"Simple" encryption: use a key to encrypt a block of data, same key will decrypt it again.

- DES - Data Encryption Standard
- AES - Advanced Encryption Standard (Rijndael)

Public Key Encryption

Keys are created as pairs - whatever one encrypts, only the other can encrypt.
First: Diffie-Hellman[2]
Common example: RSA[3] – discovered by Clifford Cocks at GCHQ in 1973, but kept classified until 1997, so Rivest, Shamir and Adleman got the credit and naming rights.

---

[2] https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange

[3] https://en.wikipedia.org/wiki/RSA_(cryptosystem)

Using Public Key Encryption

Publish one key, keep the other secret.
Encrypt using secret key – anyone can decrypt using your public key, and know you were the sender: a signature.
Anyone can encrypt using your public key – only you can decrypt the result.
Combine the two: if we both have known public keys, you can send me a message which only I can read, and only you could have sent.
It's quite slow, though – very large numbers involved – so typically, encrypt the message with a symmetric algorithm and random key, then just encrypt that key using RSA. Likewise, to sign you take a hash then encrypt that using your private key, rather than encrypting the data directly.

Certificates

Given Amazon's public key, we can send them our credit card information safely – but how do we know it's really Amazon's key we're using?
They have a certificate (X.509v3) signed by somebody our browser trusts (Symantec, as it happens).
A machine-readable statement that "www.amazon.com" belongs to "Amazon.com, Inc." of Seattle, Washington, US and uses the public key 0x3082010a..., followed by a SHA-256 hash of that statement, encrypted using Symantec's private key.
The original issuer can also revoke a certificate.

Hashing, Message Digests

Who says Symantec can issue certificates?
The CA/Browser forum, cabforum.org.

- Domain Validated:
  - "This is www.g00g1e.com"
- Organizational Validated:
  - "www.google.com belongs to Google, Inc"
- Extended Validation:
  - "online.bankofscotland.co.uk belongs to Lloyds Banking Group plc"

Client Certificates

Sometimes used for authentication – same idea as server certificates, but anyone can issue them. Microsoft uses them internally, for example for Azure's infrastructure, and Startcom uses them for client authentication.
Useful for authenticated and/or encrypted e-mail: anna@example.com can send a signed email to her friend bruce@example.com, who will know it hasn't been tampered with – then he can reply, encrypting it with her public key. Any observer will have no idea about the content of the message, but they will see who is sending messages, when they are sent and how large the messages are – the metadata. This can be useful information in itself.

SSL

By combining both forms of encryption along with message digests, we get various secure communication schemes. Notably:

- S/MIME and PGP for signed and/or encrypted email
- SSH/SCP for remote control and file transfer
- SSL/TLS for web, email

SSL/TLS is the most prominent, originally created by Netscape for web use. SSLv2 and SSLv3 are now considered obsolete and insecure, newer versions TLS v1.0, 1.1 and 1.2 are still in use.

Recap, Practical Tasks

Security is about A talking to B, without E listening in, knowing who the other is.
Lab tasks for week 1:

1. What does Let's Encrypt do, and how is it secured?
2. What is wrong with blackboard.abertay.ac.uk's SSL setup to earn it an F?
3. Why does Google only get a B from ssllabs.com for google.com's SSL setup?
4. What are: ECDHA, RSA, 3DES, AES, CHACHA20, POLY1305 and GCM in SSL?

Questions?

Any questions?

- Avoid logging in from Netlab/Hacklab machines
- Put your coursework on a memory stick or similar
- Alternatively, use Dropbox or Google Docs
- Use two factor authentication if you must log in