Class 4: Modern Cryptography

http://www.cs.virginia.edu/jillcrypto

Menu

- Some loose ends on WWII
- Maurice Burnett
- Modern Cryptography
  - Modern symmetric ciphers
  - Public-key cryptosystems

British Cipher Machine
- Design based on commercial Enigma
- 5 rotor wheels (instead of 3 in Enigma)
- Multiple rings per rotor
- Last 2 rotor wheels didn’t rotate
- British attempted to break it (without success)

Typex Machine

German Code-Breaking Efforts
- About 6,000 people (compare to 12,000 working at Bletchley Park)
- Decentralized: each military branch had their own, didn’t share what they learned
- Effective against manual codes: broke about 50% of manually coded messages
- Didn’t attempt to break rotor-based ciphers – so confident Enigma was unbreakable, didn’t try to Typex and similar machines

Lorenz Cipher

From http://www.codesandciphers.org.uk/lorenz/fish.htm
Modern Symmetric Ciphers

A billion billion is a large number, but it’s not that large a number.
Whitfield Diffie

• Same idea but:
  – Use digital logic instead of mechanical rotors
  – Larger keys (random bits, not rotor alignments)
    • Lorenz ≈ 5^{12} < 10^9
    • Modern ≥ 128 bits > 10^{37}
  – Encrypt blocks of letters at a time

Modern Ciphers

• AES (Rijndael) successor to DES selected 2001
• 128-bit keys, encrypt 128-bit blocks
• Brute force attack (around 10^{30} times harder than Lorenz)
  – Try 1 Trillion keys per second
  – Would take 1079028307086000000 years to try all keys!
  – If that’s not enough, can use 256-bit key
• No known techniques that do better than brute force search

Login Process

Terminal

| Login: alyssa | Password: fido |

login sends <"alyssa", "fido">

Trusted Subsystem

Eve

Sending Passwords

User

Server

Encrypt

Decrypt

K

Ciphertext

Plaintext

The Internet

C = Encrypt_{K}(P)
P = Decrypt_{K}(C)

Key Agreement Demo

(Animated version at end of slides.)
Asymmetric Cryptosystems

- Need a hard problem (like symmetric cryptosystems)
- With a trap door: if you know a secret, the hard problem becomes easy

One-Way Functions

- Easy to compute, hard to invert
- Trap-door one way function:
  - $D(E(M)) = M$
  - $E$ and $D$ are easy to compute.
  - Revealing $E$ doesn’t reveal an easy way to compute $D$.
  - Hence, anyone who knows $E$ can encrypt, but only someone who knows $D$ can decrypt

RSA [Rivest, Shamir, Adelman 78]

One-way function: multiplication is easy, factoring is hard
Trap-door: number theory (Euler and Fermat)

Security of RSA

- $n$ is public, but not $p$ and $q$ where $n = pq$
- How much work is factoring $n$?

Number Field Sieve (fastest known factoring algorithm) is:

$$O\left(e^{1.9223\left(\ln(n)\right)^{1/3}\left(\ln(\ln(n))\right)^{2/3}}\right)$$

$n \sim 200$ digits – would take quintillions of years

Asymmetric Cryptosystems

- Encryption and Decryption are done with different keys
- Keep one of the keys secret, reveal the other

$E_{KRA}(E_{KUA}(M)) = M$

Alice’s Public Key: KUA
Alice’s Private Key: KRA

Only KRA can decrypt a message encrypted using KUA.

Public-Key Applications: Privacy

- Alice encrypts message to Bob using Bob’s Public Key
- Only Bob knows Bob’s Private Key $\Rightarrow$ only Bob can decrypt message
Signatures

- Bob knows it was from Alice, since only Alice knows Alice’s Private Key
- Non-repudiation: Alice can’t deny signing message (except by claiming her key was stolen!)
- Integrity: Bob can’t change message (doesn’t know Alice’s Private Key)

Key Management

Approach 1: Meet Secretly
- User and Server Operator meet secretly and swap public keys
  - If you can do that, might as well agree on a secret (symmetric key) instead
  - Doesn’t work for Internet transactions

Approach 2: Public Announcement
- Publish public keys in a public forum
  - Append to email messages
  - Post on web site
  - New York Time classifieds
- Easy for rogue to pretend to be someone else
  - Forge email, alter web site, lie to New York Times

Approach 3: Public Directory
- Trusted authority maintains directory mapping names to public keys
- Entities register public keys with authority in some secure way
- Authority publishes directory
  - Print using watermarked paper, special fonts, etc.
  - Allow secure electronic access
    - Depends on secure distribution of directory’s key
Approach 4: Certificates

VeriSign

Request

Server

User

$K_U$

$E_{K_U}(C_S) = ["Server", K_U]$

Note: This is slightly simplified from the actual SSL protocol. It is vulnerable to a man-in-the-middle attack!

SSL (Secure Sockets Layer)

Browser

Server

Hello

Check Certificate using $K_{CA}$

Pick random $K$

$K_U[K]$

Secure channel using $K$

Find $K$

using $K_{RS}$

Note: This is slightly simplified from the actual SSL protocol. This version is vulnerable to a person-in-the-middle attack!

Data encrypted using secret key exchanged using some public key associated with some certificate.
SSL Recap

Browser

Check Certificate using KUCA
Pick random K

KRCA[Server Identity, KUS]

Server

KUS[K]

Find K using KRS

Secure channel using K

Note: This is slightly simplified from the actual SSL protocol. This version is vulnerable to a person-in-the-middle attack!

Questions?

Animated version of Asymmetric Cryptography Demo

Padlocked Boxes

Alice

Alice's Padlock

Alice's Padlock Key

Shady Sammy's Slimy Shipping Service

Padlocked Boxes

Alice

Alice's Padlock Key