CS164

Week 9

Symmetric Encryption

- Encryption and decryption functions: $E(x)$ and $D(x)$
- Use the same key—shared secret
- On plaintext message m sender computes ciphertext $c = E(m)$
- Receiver computes $m = D(c) = D(E(m))$
- Often we also have $m = E(D(m))$

Perfect Encryption: One-Time Pad

- Both sides share random string of bits
- Sender computes XOR of plaintext with random string
- Receiver computes XOR of ciphertext with random string
- Example:

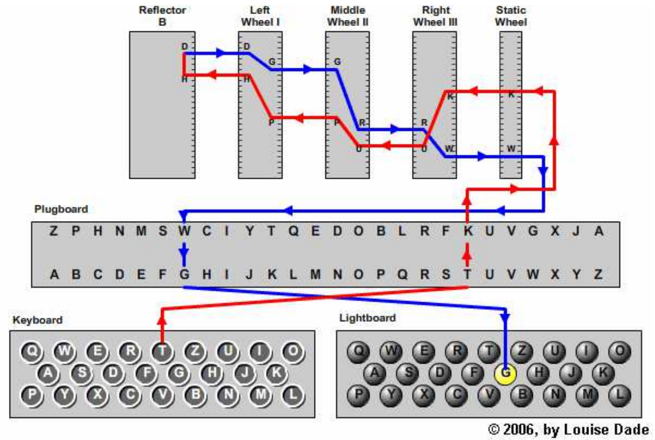
$m =$CCI	0 1 0 0 0 0 1 1 0 1 0 0 0 0 1 1 0 1 0 0 1 0 0 1
$\oplus$	0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 0
$c = E(m) =$WBW	0 1 0 1 0 1 1 1 0 1 0 0 0 0 1 0 0 1 0 1 0 1 1 1
$\oplus$	0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 0
$D(c) =$CCI	0 1 0 0 0 0 1 1 0 1 0 0 0 0 1 1 0 1 0 0 1 0 0 1

Symmetric Encryption Standards

- Data Encryption Standard (DES)
- Advanced Encryption Standard (AES)
- International Data Encryption Algorithm (IDEA)
- RC4 used in:
  - Secure Sockets Layer (SSL)
  - Wired Equivalent Privacy (WEP)

Enigma Machine

Enigma Machine



Modular Arithmetic

- Arithmetic on a circle
  - E.g. A clock: 10 o'clock + 3 hours = 1 o'clock
  - 13 is congruent to 1 modulo (mod) 12
  - Less formally: 13 mod 12 is 1
- As an operator, mod is basically the remainder
- Lots of interesting properties we don't have time to go into

Relative Primeness

- Prime: n is prime if it has no integral divisors other than 1 and itself
- Relatively prime: n and m are relatively prime if they share no integral divisor other than 1
- I.e. the greatest common divisor of n and m is 1
- $\gcd(n, m) = 1$

GCD Algorithm

1. Compute r as the remainder when m is divided by n
2. If $r = 0$, stop and output the value of n as the GCD.
3. Otherwise :
   (a) replace m by the value of n
   (b) replace n by the value of r
   (c) return to Step 1

Diffie-Hellman Key Exchange

- Public key technique to establish a shared secret without ever transmitting the secret
- Two numbers, g and p where p is prime are publically known
  1. Alice generates random number a and Bob generates random number b
  2. Alice transmits $g^a \bmod p$ to Bob and Bob transmits $g^b \bmod p$ to Alice
  3. Alice and Bob compute $(g^b)^a \bmod p = g^{(ab)} \bmod p = (g^a)^b \bmod p$

Public Key Encryption

- Assymetric: different keys for encryption and decryption
- No shared secret
- Encryption key is public and decryption key is private
- Anyone can encrypt a message for anyone else, but only the intended recipient can read it

RSA Key Generation

1. Pick large random numbers p and q
2. Let $n = pq$
3. Compute $\phi(n) = (p-1)(q-1)$
4. Pick e relatively prime to $\phi(n)$
5. Find d such that $ed = 1 \bmod \phi(n)$
6. Publish e and n; d is kept private
7. $E(x) = x^e \bmod n$
8. $D(x) = x^d \bmod n$
9. $x = E(D(x)) = D(E(x)) = x^{ed} \bmod n$

Signatures

- In PKC, how do we know the sender is real?
- Answer: append a signature that can only come from the purported sender:
  1. Alice (a) is sending to Bob (b)
  2. Alice computes $c = E_b(m)$
  3. Alice computes $s = H(m)$ where H is a cryptographic hash function
  4. Alice sends $(c, D_a(s))$
  5. Bob verifies that $H(D_b(c)) = E_a(D_a(s))$
  6. Only Bob can read c and only Alice could have sent $D_a(s)$

Certificates

- How does a sender know it has the right public key for a recipient?
- Answer: a certificate from a mutually trusted party called a certifying authority (CA)
- CA answers queries with a certificate containing the public key in question and a signature from the CA