Public Key Cryptography

BİL 448/548
Internet Security Protocols
Ali Aydınlı Selçuk
Public Key Cryptography

- The single most important idea in modern cryptography.
- Proposed by Diffie & Hellman, 1976.
- Asymmetric key cryptography:

\[
P = D_K(C) \quad \text{plaintext}
\]

\[
C = E_{K'}(P) \quad \text{ciphertext}
\]

\[
\text{pub. key } K' \quad \text{prv. key } K
\]

- It shouldn’t be possible to obtain \( K \) from \( K' \). So, \( K' \) can safely be made public.
PKC solves the classical “key distribution problem”:  
– If there is no secure channel, how can A & B share the key securely?

PKC solution:  
– Alice makes her encryption key K' public  
– Everyone can send her an encrypted message:  
  \[ C = E_{K'}(P) \]  
– Only Alice can decrypt it with the private key K:  
  \[ P = D_K(C) \]
PKC also solves the message source authentication problem:

- Only Alice can “sign” a message, using K.
- Anyone can verify the signature, using K'.

Only if such a function could be found...
Discrete Logarithm Problem

- DLP: Given \( g \) and \( y = g^x \), what is \( x \)?

- Easy over \( \mathbb{Z} \).
  E.g., if \( 2^x = 4096 \), \( x = 12 \).

- Hard over \( \mathbb{Z}_p \).
  E.g., if \( 2^x = 28 \) (mod 113), \( x = ? \)
Diffie-Hellman Key Exchange

- Public: prime $p$, generator $g$.
- Alice chooses random $a$ (secret); Bob chooses random $b$ (secret).

Alice

$g^a \mod p$

$g^b \mod p$

Bob

$K = g^{ab} \mod p$

$K = (g^b)^a \mod p$

$K = (g^a)^b \mod p$
Security of DH

• **Discrete Log Problem:** Given $p$, $g$, $g^a \mod p$, what is $a$?

• **DH Problem:** Given $p$, $g$, $g^a \mod p$, $g^b \mod p$, what is $g^{ab} \mod p$?

• **Conjecture:** DHP is as hard as DLP.
  (note: Neither is proven to be NP-hard.)
Efficiency of DH

Generating a large prime

- Generate a random number & test for primality.
- Primality testing is efficient.
- Density of primes:

**Prime Number Theorem:** For $\pi(n)$ denoting the number of primes $\leq n$, we have

$$\pi(n) \sim \frac{n}{\ln n}.$$

That is,

$$\lim_{n \to \infty} \frac{(\pi(n) \ln n)}{n} = 1.$$
Efficiency of DH

How to compute \((g^a \mod p)\) for large \(p, g, a\)?

\[
x^n = \begin{cases} (x^k)^2 & \text{if } n = 2k \\ (x^k)^2x & \text{if } n = 2k + 1 \end{cases}
\]

“Repeated squaring”: Start with the most significant bit of the exponent.

E.g. Computing \(3^{25} \mod 20\). \(25 = (11001)_2\)

\[
\begin{align*}
y_0 &= 3^{(1)} \mod 20 = 3 \\
y_1 &= 3^{(11)} \mod 20 = 3^2 \cdot 3 \mod 20 = 7 \\
y_2 &= 3^{(110)} \mod 20 = 7^2 \mod 20 = 9 \\
y_3 &= 3^{(1100)} \mod 20 = 9^2 \mod 20 = 1 \\
y_4 &= 3^{(11001)} \mod 20 = 1^2 \cdot 3 \mod 20 = 3
\end{align*}
\]

Further efficiency with preprocessing \(x^i, i < 2^k\), for some \(k\).