Public Key Cryptography

- Use two different keys for encryption and decryption
  - Classical cryptography requires a common key
  - Public key cryptography not

Public and Private Keys

- An entity has two keys: a **public key** and a **private key**
- Its public key is public
  - Everybody is assumed to know it
- Its private key is secret
  - Nobody is assumed to know it
- Hard to derive the private key from the public key

Properties of Public Key

- Assuming x has a public key \( U \) and a private key \( R \)
- Message encrypted with \( U \) can **only** be decrypted by x using \( R \)
  - Useful to send an encrypted message to x
- If a message can be decrypted with \( U \), then it must be encrypted by x using \( R \)
  - Useful to verify whether or not a message is from x

Public Key Cryptosystems

- Diffie-Hellman
- RSA
- . . . . .

Diffie-Hellman

- First public key cryptosystem
  - Still in use today
- Used to generate a **common** key by two users
Discrete Logarithm Problem

- Find a value of \( k \) such that
  \[ K = g^k \mod p \]
  for a given \( K, g \), and prime \( p \).
- Difficulty increases exponentially as \( p \) increases
- This is the basis of Diffie-Hellman

Algorithm

- All users share \( p \) and \( g \)
- Each user \( u \) chooses a private key \( k(u) \) and a public key \( K(u) \)
- \( K(u) = g^{k(u)} \mod p \)
- When users A and B communicate,
  \[ A: \ s(A) = F_{k(A)}(K(B)) = K(B)^{k(A)} \mod p \]
  \[ B: \ s(B) = F_{k(B)}(K(A)) = K(A)^{k(B)} \mod p \]
  \( s \) will be used as the secret key for A-B comm.

Example

- Alice and Bob chose \( p = 53, g = 17 \)
- \( k(Alice) = 5, k(Bob) = 7 \)
- \( K(Alice) = 17^5 \mod 53 = 40 \)
- \( K(Bob) = 17^7 \mod 53 = 6 \)
- Alice: \( K(Bob)^{k(Alice)} \mod p = 6^5 \mod 53 = 38 \)
- Bob: \( K(Alice)^{k(Bob)} \mod p = 40^7 \mod 53 = 38 \)

Diffie-Hellman Summary

- Based on the computational infeasibility to derive the private key of a public key
  - \( p \) must be very large (hundreds of bits)
- Diffie-Hellman is an example of symmetric key exchange protocol

RSA

- An exponentiation cipher
- Choose two large prime numbers \( p \) and \( q \)
- \( n = pq \)
- \( Q(n) = \{ x | x < n, \text{and} (x, n) = 1 \} \)
- Totient \( \phi(n) \): \( | Q(n) | \)

Examples

- Let’s just use small numbers to illustrate
- Say \( n = 10 \)
- \( Q(10) = \{ 1, 3, 7, 9 \} \)
- \( \phi(10) = 4 \)
RSA public and private keys

• Public key
  – An integer $e < n$ that is relatively prime to $\phi(n)$

• Private key:
  – An integer $d$ that satisfies $ed \mod \phi(n) = 1$

RSA Encryption and Decryption

• Encryption ($m$ is plaintext, $c$ is ciphertext)
  \[ c = m^e \mod n \]

• Decryption
  \[ m = c^d \mod n \]

Example

• Let $p = 7$, $q = 11$
• $n = 77$
• $\phi(77) = 60 = |\{1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 13, \ldots \}|$
• Choose $e = 17$ as public key, $d = 53$ as private key

Example (cont’d)

• Use number 0 to 25 to represent A to Z

• Confidentiality
  – Use the recipient’s public key to encrypt
  – Letter “H” becomes 7$^ {17}$ mod 77 = 28
  – “HELLO WORLD” becomes “28 16 44 44 42 38 22 42 19 44 75”

• Origin authentication
  – Use the sender’s private key to encrypt
  – Letter “H” becomes 7$^ {53}$ mod 77 = 35
  – “HELLO WORLD” becomes “35 09 44 44 93 12 24 94 04 05”
  – Only decryptable with the sender’s public key

More on Origin Authentication

• Also called nonrepudiation

• If a message can be deciphered with A’s public key
  – Then it must have been encrypted using A’s private key, not other’s
  – Only A knows A’s private key (assuming not stolen)
  – So it must be from A

Combine Confidentiality and Authentication

• For confidentiality, the message has to be encrypted with B’s public key
  – So that B’s private key has to be used to decrypt
  – But only B knows B’s private key

• For origin authentication, the message has to be encrypted with A’s private key
  – So that A’s public key has to be used to decrypt
  – Everybody knows A’s public key

• Question: can we switch the two above?
Cryptographic Checksums

- Motivating question: How can Bob verify messages received from Alice is not changed?
- Answer: digital signature
  - Which relies on cryptographic checksum function
  - Digital signature will be covered later
- Cryptographic checksum function also has many other usages
  - Such as S/Key protocol (covered in Authentication later)

Cryptographic Checksum Function

- Also called strong hash function
  - Or strong one-way function
- \( h: A \rightarrow B \)
  - For any \( x \in A \), \( h(x) \) is easy to compute
  - For any \( y \in B \), computationally infeasible to find \( x \in A \) such that \( h(x) = y \)
  - No collision pairs.

Prevention of Collision Pairs

- Statement A:
  - Computationally infeasible to find \( x, x' \in A \) such that \( x \neq x' \) but \( h(x) = h(x') \)
- Statement B:
  - Given any \( x \in A \), computationally infeasible to find another \( x' \in A \) such that \( x \neq x' \) but \( h(x) = h(x') \)
- Statement B is much harder than statement A.

Keyed/Keyless Cryptographic Checksum

- A keyed cryptographic checksum requires a cryptographic key as part of hashing computation
  - E.g. DES-MAC (DES in CBC mode)
  - Use last block output as the hash result
  - DES needs a key
- A keyless cryptographic checksum does not
  - MD2, MD4, MD5
  - SHA-1 (Secure Hash Algorithm)
  - Snefru
  - HAVAL

HMAC

- An algorithm to produce a keyed hash function from a keyless hash function and a key.
- Read 8.4.1 in textbook.