UC

Public-Key Infrastructure (P

Junghoo Cho

cho@cs.ucla.edu

Public-Key Infrastructure (PKI) - Junghoo Cho - cho@cs.ucla.edu

Four Security Guarantees

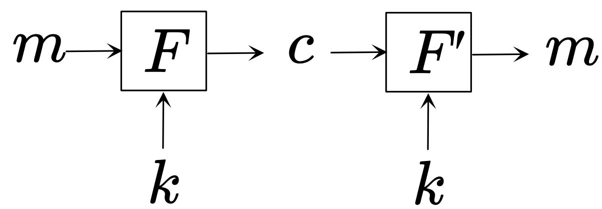
- Internet is an open and public forum where everyone talks t else
  - Data packets can be intercepted and seen by anyone
  - No guarantee on the origin and integrity of data packet
- Q: Given this, what guarantees may we desire before we con important transactions over the Internet?
  1. *Confidentiality*
  2. *Message/data integrity*
  3. *Authentication*
  4. *Authorization*

Confidentiality

- Q: How can we keep confidentiality of the messages?

1. *Steganography*: “embed” true message within harmless-looking mes
    - Kathy is laughing loudly
    - Change the lowest bit of image pixels
    - “Security by obscurity”
2. *Encryption*: “scramble” message with a secret key, so that it wouldn’t to others unless they have the key
    - Example: bitwise XOR with k
    - 11110000 (message) XOR 10111001 (key) → 01001001 (ciphertext)
    - 01001001 (ciphertext) XOR 10111001 (key) → 11110000 (message)

Symmetric-Key Cipher



- $F(m, k)$: encryption function, e.g., $F(m, k) = m$ XOR k
  - m: plaintext (= message), k: secret key
  - c: ciphertext. transmitted over insecure channel
- $F'(c, k)$: decryption function, e.g., $F'(c, k) = c$ XOR k
  - Inverse of F: $F' \cdot F = I$
- The pair $[F(m, k), F'(c, k)]$ is called a *cipher*

Security of Cipher

- Q: What property should $F(m, k)$ have?
- A: Ideally, one should never be able to guess m from c alone
  - Ciphertext should not reveal any information about plaintext
- *Perfect secrecy* (= *Shannon secrecy*)
  - For all plaintext x and ciphertext y, $Pr(x|y) = Pr(x)$
- OTP (one time pad) encryption is proven to be perfectly secr due to practical limitation, cannot be used directly
  - Many encryption algorithms try to “mimic” OTP, e.g., RC4

Popular Ciphers

- AES (advanced encryption standard)
  - 128 bit block cipher
  - 128, 192, 256 bit keys
  - Adopted by NIST (national institute of standard and technology) as replacement of DES in 2000
- IDEA, A5 (used by GSM), …

Challenges

- Q: Can A use the same key for communicating with B and C
- Q: If there are n parties, how many keys are needed?
- Q: How can two parties agree on a key "secretly" over the Int
  the first place?

Key Agreement Problem

- Q: Can two parties send and receive encrypted messages wi
agreeing on a shared secret key?
- A: *Asymmetric-key cipher*

Asymmetric-Key Cipher

- Two pairs of keys, not one!
  - e: encryption key
  - d: decryption key
- Q: How does this help?

Asymmetric-Key Cipher

- Everyone has their own (e, d) key pair
- Everyone shares their e with anyone: *public key*
  - Other users use the public key to encrypt a message to the user
- Users keep their d secret: *private key*
  - Users use their private key to decrypt message
- No need to send the private key over insecure channel
  - Private key *NEVER* leaves the owner of the key

Asymmetric-Key Cipher

- Idea first developed by Ellis, Cocks, and Williams (working for NSA)
  - In early 70's, but could not publish
- First public-key cryptosystem by Diffie and Hellman in 1976
- RSA (Rivest, Shamir and Adleman)
  - Most widely used asymmetric-key cipher
  - Used by many security protocols: SSL, PGP, CDPD, …

RSA: Key Generation

1. Pick two *random* prime numbers p and q.
2. Pick $e < (p-1)(q-1)$
   - e does not have to be random
   - Popular choice: $e = 65537 (= 2^{16}+1), 3, 5, 35, \ldots$
3. Find $d < (p-1)(q-1)$ such that $de \bmod (p-1)(q-1) =$
   - Using *extended-euclid algorithm*
4. (e, n) becomes public key, (d, n) becomes private key where
   - Throw away p and q

RSA Cipher

- Encryption and Decryption functions
  - $F(m, (e, n)) = m^e \bmod n$
  - $F'(c, (d, n)) = c^d \bmod n$

$$m \longrightarrow \underset{\substack{\uparrow \\ (e,\, n)}}{\boxed{m^e \bmod n}} \longrightarrow c \longrightarrow \underset{\substack{\uparrow \\ (d,\, n)}}{\boxed{c^d \bmod n}} \longrightarrow m$$

- Q: Does this work?

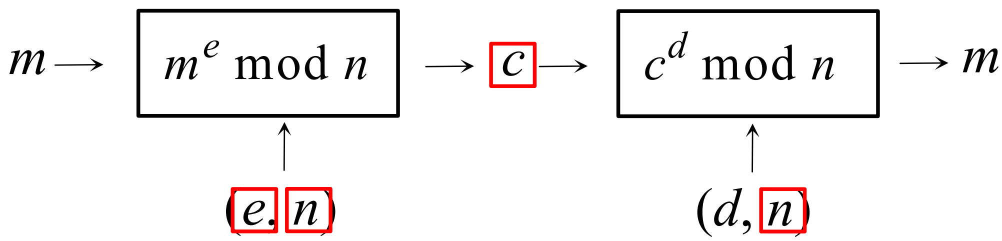
RSA: Two Important Theorems

- Q: Given a choice of e, can we always find d such that $de \bmod$ $1)(q-1) = 1$?
- A: Yes, there exists unique d if e is a *coprime* to $(p-1)(q-1$
  - i.e., e does not share any factor with $(p-1)(q-1)$
- Q: Is $F'(c,(d,n))$ the inverse of $F(m,(e,n))$?
- A: Yes, $m = [(m^e \bmod n)^d \bmod n]$ for such e, d and $n = pq$
- RSA works!
  - But most asymmetric-key ciphers are 1000x slower than any symme cipher
- Q: Is it secure? What should we make sure for the security of

Security of Asymmetric-Key Cipher

- Q: What properties should F, F', e, and d satisfy to make thi
- A: One should never guess m from c without d (~ perfect sec
- A: One should never guess d from e

Security of RSA (1)



- Q: Can a hacker "break RSA"?
- Q: What does the hacker know? m? c? (e, n)? (d, n)?
- Q: What other relationship does the hacker know?
- A: $de \bmod (p-1)(q-1) = 1, \quad n = pq, \quad c = m^e \bmod n$

Security of RSA (2)

$$de \bmod (p-1)(q-1) = 1, \quad n = pq, \quad c = m^e \bmod n$$

- Q: Can the hacker get m by solving $c = m^e \bmod n$?
- A: *RSA problem*. No efficient solution known.
- Q: Can the hacker get d by solving $de \bmod (p-1)(q-1) =$
- Q: Can the hacker get p and q from $n = pq$?
- A: *Large-number factorization problem*. No efficient solution k

Security of RSA (3)

- Security of RSA depends on the difficulty of two key problen
  - RSA problem: solve $c = m^e \bmod n$ for m
  - Large-number factorization problem: factorize $n = pq$ for large n, p

Application of Asymmetric-Key Cipher

- Q: How can we use an asymmetric-key cipher to keep messa "confidential"?
- A:
  1. Use asymmetric-key cipher to establish a shared key
  2. Using the shared key, use symmetric-key cipher to encrypt message
     - Performance and complexity issue
- Q: How can we "authenticate" the other party?
- A: Challenge-Response
  - Challenge: generate random value r and send $c = F(r, e)$
  - Response: send back $F'(c, d) = r$
  - Only the one with d can send back r

Application of Asymmetric-Key Cipher

- Q: How can we check the message integrity? How can we ma
  others did not temper with message?
- A: *Signature*
  - Main idea: $I = F' \cdot F = F \cdot F'$. That is, $F(F'(m,d),e) = m$!
    - In RSA, for example, $m = (m^e \bmod n)^d \bmod n = (m^d \bmod n)^e \bmod n$
  - "Private-key decrypted" checksum of message body
  - Given a message with signature, "encrypt" the signature using the p
    the author
  - Correct signature should have correct checksum

Public-Key Infrastructure

- Q: How do we know the public key for A *really* belongs to A?
- Q: In real world, how do we verify the identity of a person?
- Q: Why do we trust it?
- A: Public-Key Infrastructure (PKI)
  - *Certificate Authority* (CA)
    - Trusted entity that can issue trusted *certificates* to Web sites
    - Performs out-of-band identity verification before issuing a certificate
  - *Certificate*
    - Text (XXXX is the public key of A) signed by CA's secret key
    - Others can "trust" the public key if they trust CA

HTTPS: High-Level Description

1. When contacted by client, server presents its signed certifica
   - “XXX is the public key of amazon.com. This certificate is valid until …
2. Client “authenticates” server through challenge/response us public key
3. Client/server agrees on a symmetric-key through a secure ch established through asymmetric-key cipher
4. Client/server communicate securely through symmetric-key

Multi-Factor Authentication

- Q: What if the user loses their secret password?
- *Multi-factor authentication*
  - To minimize possibility of compromised keys, systems authenticate on combinations of
    - What you have (e.g., physical key, id card)
    - What you know (e.g., password)
    - Who you are (e.g., fingerprint)
  - *2-factor authentication*

Popular Second Factor

- Smartphone
  - Send an SMS/push notification on a registered device
- USB key
  - e.g., FIDO U2F Security Key

- SmartCard
  - Temper-resistant security card

Popular Second Factor

- OTP (one time password) key
  - A physical card flashing a new security code, say, every minute
    - e.g. SecurID by RSA security
  - User provides the security code to log in

What We Learned

- Four security guarantees
  - Confidentiality, integrity, authentication, authorization
- Symmetric-key cipher: AES algorithm
- Asymmetric-key cipher: RSA algorithm
- Public-Key Infrastructure (PKI)
  - Certificate Authority (CA), certificate
- HTTPS
- Multi-factor authentication