Network Security (I)

Security

- Security is a broad problem. In this course, we will look at securing communication protocols.
- In the previous classes, we mentioned that when you connect your laptop to public WiFi and visit your bank's website, we need HTTPS to prevent someone from stealing my bank information from the transferred packets, as it encrypts the packet contents making it hard to interpret. But, is encryption sufficient enough? What could go wrong? [the bank website may not be the actual bank website; your transaction may be altered; or your deposition could be repudiated]
- Objectives
  - confidentiality
    - realized via encryption
    - the content of the transmitted message should only be understood by the sender and receiver
    - an intercepted message cannot be understood by any interceptor/eavesdroppers
  - authentication
    - confirm the other party involved in the communication is indeed who or what they claim to be
    - no simple if we cannot communicate with the other party face-to-face
  - message integrity
    - make sure the content of the communication is not altered
    - checksumming techniques can be used to provide message integrity
  - non-repudiation
    - Non-repudiation is the concept of ensuring that a party in a dispute cannot repudiate, or refute the validity of a statement or contact (digital signatures)

Cryptography

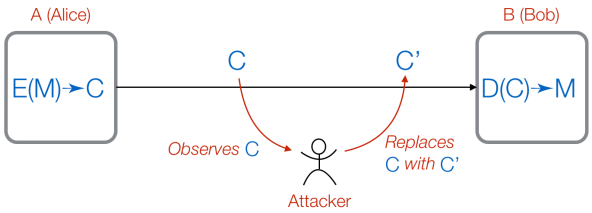
- Cryptographic techniques are relevant to all four objectives we described above.
- Long history since at least Julius Caesar
- Modern cryptographic techniques, including many of those used in the Internet, are based on the advances made in the past 30 years
- We only touch the essential aspects of cryptography, particular as they are practiced on the Internet.

Encryption



- Alice disguises the message/plaintext (M) by using an encryption algorithm (E).
- The encrypted message/ciphertext (C) looks unintelligible to any intruder.
- Bob receives the ciphertext and decrypted it by some decryption algorithm (D).
- Question: If an attacker cannot read an encrypted message, can they still change it? [Potentially. Encryption alone does not necessarily guarantee integrity.]

Encryption - Attacks



- Passive attack: message observed
- Active attack: message replaced or modified

- Generally speaking, the attacks can be categorized into two groups:
  - Passive attack: intruders passively observe the cipher text, decrypt the cipher text, and then know the transmitted messages.
  - Active attack: intruders intercept the cipher text and replace with another message or modify the intercepted message and then send it to B.

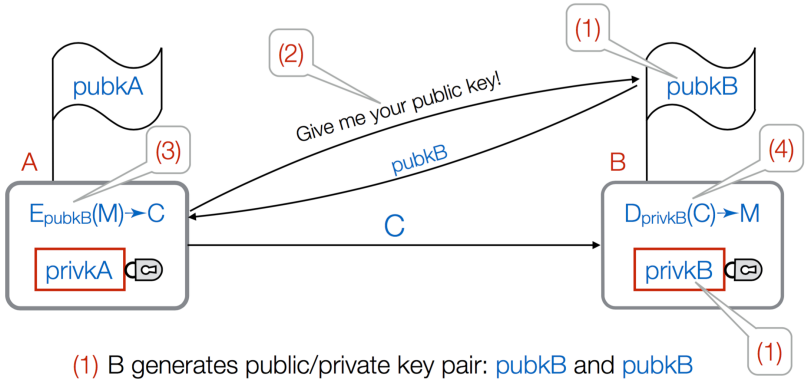
Encryption Categories

- Secret method: E( ) and D( )
  - Encrypt message using a secrete method known by Alice and Bob only
  - Not quite feasible in Internet, as the encryption technique is available to everyone.
    - These methods are published and standardized. E.g. MD5/RFC1321, Public-Key Cryptography Standards/RFC3447, PPP Triple-DES Encryption Protocol/RFC2420.
  - So, there must be some secrete information that prevents an intruder from decrypting. Therefore, the secret key is introduced.

- Public method, secret key: $E_k( )$ and $D_k( )$
  - Encryption algorithm is public, but the key is private, which are known by Alice and Bob only.
  - So, even if intruders sniffer the message, they cannot interpret it without the secrete key.
  - Since the sender and receiver use the same private key to encrypt messages, this type of methods are called Symmetric Key Protocols:
    - DES - Data Encryption Standard

    - 1977, symmetric key, 56-bit key, 64-bit data
  - AES - Advanced Encryption Standard
    - 1998, symmetric key, 128, 192, and 256-bit keys, 128-bit data blocks
- However, this requires that Alice and Bob have an agreement on the private key.
  - They can first meet and agree on the key in person.
  - But, in Internet this is infeasible. Agreement on the key must be made over another secure communication. This secure communication faces the same problem.

- Public method, public and private keys: $E_{pubk}( )$ and $D_{privk}( )$
  - To address the issue of the agreement on the private, each person has two keys: a public key and a private key
  - Public key is accessible to everyone, including intruders. Private key is accessible to the receiver only. The public key and private key are carefully selected and are paired.
  - Alice uses the Bob's public key along with a standard encryption algorithm to send a message to Bob
  - After Bob receives the message, he uses his private key to decrypt the ciphertext.



(1) B generates public/private key pair: pubkB and pubkB
(2) A gets B's public key
(3) A encrypts the message: $E_{pubkB}(M) \rightarrow C$ and sends it to B
(4) B decrypts the message: $D_{privkB}(C) \rightarrow M$

RSA - Rivest, Shamir, and Adleman

- 1978, public/private key algorithm, 1,024 to 4,096-bit keys (typically)

- Requirements:
  - need $K_B^+(.)$ and $K_B^-(.)$ such that $K_B^-(K_B^+(m)) = m$
  - given public key K_B^+, it should be impossible to compute (which is why key is 1024 to 4096 bits) private key K_B^-

- RSA: getting ready
  - message: just a bit pattern
  - bit pattern can be uniquely represented by an integer number

- thus, encrypting a message is equivalent to encrypting a number
- Example:
  - `m = 1001 0001`. This message is uniquely represented by the decimal number 145.
  - to encrypt `m`, we encrypt the corresponding number, which gives a new number (the ciphertext).

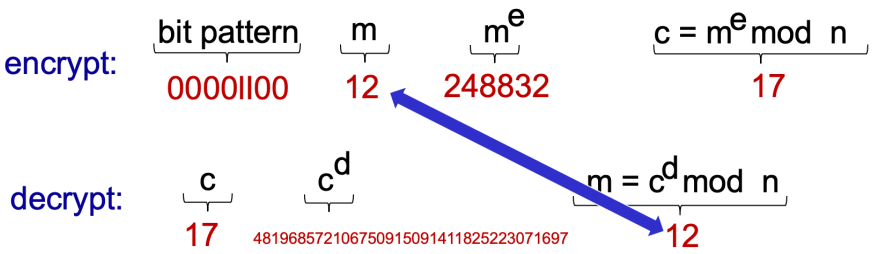
- RSA: Creating public/private key pair
  1. choose two large prime numbers `p, q`. (e.g., 1024 bits each)
  2. compute `n = pq, z = (p-1)(q-1)`
  3. choose `e` (with `e < n`) that has no common factors with `z` (`e, z` are "relatively prime").
  4. choose `d` such that `ed-1` is exactly divisible by `z`. (in other words: `ed mod z = 1`).
  5. public key K_B^+ is `(n, e)`. private key K_B^- is `(n, d)`.

- RSA: encryption, decryption
  - given `(n, e)` and `(n, d)` as computed above
  - to encrypt message `m (m < n)`, compute $c = m^e \bmod n$
  - to decrypt received bit pattern, `c`, compute $m = c^d \bmod n$

magic happens! $m = \underbrace{(m^e \bmod n)}_{c}{}^d \bmod n$

from number theory [Kaufman 1995]

- RSA example:
  - B choose `p = 5, q = 7`, Then `n = 35, z = 24. e = 5` (so, `e, z` relatively prime). `d = 29` (so `ed - 1` exactly divisible by `z`).
  - encrypting 8-bit messages.



- Is RSA secure enough?
  - suppose you know B's public key (n, e). How hard is it to determine d?
  - essentially need to find factors of n without knowing the two factors p and q
    - fact: factoring a big number is hard
  - How do you break it?

    - if there is algorithms for quickly factoring a number [not known yet, but be Quantum Computer]

- RSA in practices: used for session keys
  - exponentiation in RSA is computationally intensive
  - DES is at least 100 times faster than RSA
  - use public/private key crypto to establish secure connection, then establish second key - symmetric session key - for encrypting data

- Does HTTPS use RSA?
  - TLS 1.0 - 1.2 use RSA. TLS 1.3 uses RSA for server authentication and other hybrid cryptographic approaches for establishing shared session secrets and encrypting application data.

Disclaimer: Notes adapted from the textbook and online resources.