Network Security Principles, Symmetric Key Cryptography, Public Key Cryptography

Modified by Xiuzhen Cheng
Originally provided by Professor Rick Han (rhan@cs.colorado.edu) at the University of Colorado at Boulder

Network Security
- Classic properties of secure systems:
  - Confidentiality
    - Encrypt message so only sender and receiver can understand it.
  - Authentication
    - Both sender and receiver need to verify the identity of the other party in a communication: are you really who you claim to be?
  - Authorization
    - Does a party with a verified identity have permission to access (r/w/x/…) information? Gets into access control policies.

Network Security (2)
- Classic properties of secure systems: (cont.)
  - Integrity
    - During a communication, can both sender and receiver detect whether a message has been altered?
  - Non-Repudiation
    - Originator of a communication can't deny later that the communication never took place
  - Availability
    - Guaranteeing access to legitimate users. Prevention of Denial-of-Service (DOS) attacks.

Cryptography
- Encryption algorithm also called a cipher
- Cryptography has evolved so that modern encryption and decryption use secret keys
  - Only have to protect the keys! => Key distribution problem
- Cryptographic algorithms can be openly published

Cryptography (2)
- Cryptography throughout history:
  - Julius Caesar cipher: replaced each character by a character cyclically shifted to the left. Weakness?
    - Easy to attack by looking at frequency of characters
  - Mary Queen of Scots: put to death for treason after Queen Elizabeth's I's spymaster cracked her encryption code
  - WWII: Allies break German Enigma code and Japanese naval code
    - Enigma code machine (right)

Cryptography (3)
- Cryptanalysis - Type of attacks:
  - Brute force: try every key
  - Ciphertext-only attack:
    - Attacker knows ciphertext of several messages encrypted with same key (but doesn’t know plaintext).
    - Possible to recover plaintext (also possible to deduce key) by looking at frequency of ciphertext letters
  - Known-plaintext attack:
    - Attacker observes pairs of plaintext/ciphertext encrypted with same key.
    - Possible to deduce key and/or devise algorithm to decrypt ciphertext.
Cryptography (4)

- Cryptanalysis – Type of attacks:
  - Chosen-plaintext attack:
    - Attacker can choose the plaintext and look at the paired ciphertext.
    - Attacker has more control than known-plaintext attack and may be able to gain more info about key.
  - Adaptive Chosen-plaintext attack:
    - Attacker chooses a series of plaintexts, basing the next plaintext on the result of previous encryption.
    - Differential cryptanalysis – very powerful attacking tool.
    - But DES is resistant to it.
- Cryptanalysis attacks often exploit the redundancy of natural language.
  - Lossless compression before encryption removes redundancy.

Principles of Confusion and Diffusion (2)

- "Confusion": a classical Substitution Cipher

- "Diffusion": a classical Transposition Cipher

Symmetric-Key Cryptography

- Both sender and receiver keys are the same: $K_A = K_B$.
- The keys must be kept secret and securely distributed – we’ll study this later.
- Thus, also called "Secret Key Cryptography".
- Data Encryption Standard (DES)
Symmetric-Key Cryptography (3)
- Data Encryption Standard (DES)
  - Encodes plaintext in 64-bit chunks using a 64-bit key (56 bits + 8 bits parity)
  - Uses a combination of diffusion and confusion to achieve security
    - $abc \rightarrow dbac$
  - Was cracked in 1997
  - Parallel attack – exhaustively search key space
  - Triple-DES: put the output of DES back as input into DES again with a different key, loop again: $3 \times 56 = 168$ bit key
  - Decryption in DES – it's symmetric! Use $K_a$ again as input and then the same keys except in reverse order
  - Advanced Encryption Standard (AES) successor

Symmetric-Key Cryptography (4)
- DES is an example of a block cipher
  - Divide input bit stream into $n$-bit sections, encrypt only that section, no dependency/history between sections

Symmetric-Key Cryptography (5)
- Electronic Code Book (ECB) mode for block ciphers of a long digital sequence
  - Vulnerable to replay attacks: if an attacker thinks block $C_2$ corresponds to $\$ amount, then substitute another $C_k$
  - Attacker can also build a codebook of $\langle C_k, guessed\ P_k \rangle$ pairs

Symmetric-Key Cryptography (6)
- Cipher Block Chaining (CBC) mode for block ciphers
  - Inhibits replay attacks and codebook building: identical input plaintext $P_i = P_k$ won't result in same output code due to memory-based chaining
  - IV = Initialization Vector – use only once

Symmetric-Key Cryptography (7)
- Stream ciphers
  - Rather than divide bit stream into discrete blocks, as block ciphers do, XOR each bit of your plaintext continuous stream with a bit from a pseudo-random sequence
  - At receiver, use same symmetric key, XOR again to extract plaintext

Symmetric-Key Cryptography (8)
- RC4 stream cipher by Ron Rivest of RSA Data Security Inc. - used in 802.11b’s security
- Block ciphers vs. stream ciphers
  - Stream ciphers work at bit-level and were originally implemented in hardware => fast!
  - Block ciphers work at word-level and were originally implemented in software => not as fast
  - Error in a stream cipher only affects one bit
  - Error in a block cipher in CBC mode affects two blocks
  - Distinction is blurring:
    - Stream ciphers can be efficiently implemented in software
    - Block ciphers getting faster
Symmetric-Key Cryptography (9)

- Symmetric key is propagated to both endpoints \( A \) & \( B \) via Diffie-Hellman key exchange algorithm
  - \( A \) & \( B \) agree on a large prime modulus \( n \), a "primitive element" \( g \), and a one-way function \( f(x) = g^x \mod n \)
  - \( n \) and \( g \) are publicly known
  - \( A \) chooses a large random int \( a \) and sends \( B \) \( AA = g^a \mod n \)
  - \( B \) chooses a large random int \( b \) and sends \( A \) \( BB = g^b \mod n \)
  - \( A \) & \( B \) compute secret key \( S = g^{ab} \mod n \)
  - Since \( x = f^{-1}(y) \) is difficult to compute, then observer who knows \( AA, BB, n, g \) and \( f \) will not be able to deduce the product \( ab \) and hence \( S \) is secure.

Symmetric Key Distribution

- Key distribution
  - Public key via trusted Certificate Authorities
      - Symmetric key?
        - Diffie-Hellman Key Exchange
        - Public key, then secret key (e.g. SSL)
        - Symmetric Key distribution via a KDC (Key Distribution Center)

Symmetric Key Distribution (2)

- Symmetric Key distribution via a KDC (Key Distribution Center)
  - KDC is a server (trusted 3rd party) sharing a different symmetric key with each registered user
  - Alice wants to talk with Bob, and sends encrypted request to KDC, \( K_{A \to KDC}(Alice,Bob) \)
  - KDC generates a one-time shared secret key \( R1 \)
    - KDC encrypts Alice’s identity and \( R1 \) with Bob’s secret key, let \( m = K_{A\to KDC}(Alice,R1) \)
    - KDC sends to Alice both \( R1 \) and \( m \), encrypted with Alice’s key, i.e. \( K_{B\to KDC}(R1, K_{A\to KDC}(Alice,R1)) \)
    - Alice decrypts message, extracting \( R1 \) and \( m \). Alice sends \( m \) to Bob.
    - Bob decrypts \( m \) and now has the session key \( R1 \)

Symmetric Key Distribution (3)

- Kerberos authentication basically follows this KDC trusted 3rd party approach
  - In Kerberos, the message \( m \) is called a ticket and has an expiration time

Public-Key Cryptography

- For over 2000 years, from Caesar to 1970s, encrypted communication required both sides to share a common secret key => key distribution problems!
- Diffie and Hellman in 1976 invented asymmetric public key cryptography - elegant, revolutionary!
  - Sender’s key differs from receiver’s key
  - Simplifies key distribution - just protect \( K_{private} \)
  - Useful for authentication as well as encryption

Public-Key Cryptography (2)

- Host (receiver) who wants data sent to it in encrypted fashion advertises a public encryption key \( K_{public} \)
  - Sender encrypts with public key
  - Receiver decrypts with private key
Public-Key Cryptography (3)

- Decryption algorithm has the property that only a private key $K_{\text{private}}$ can decrypt the ciphertext, and it is computationally infeasible to deduce $K_{\text{private}}$ even though the attacker knows the public key $K_{\text{public}}$ and the encryption algorithm.

Public-Key Cryptography (4)

- Decryption algorithm has the property that only a private key $K_{\text{private}}$ can decrypt the ciphertext.
- Based on the difficulty of factoring the product of two large prime numbers.
- Example: RSA algorithm (Rivest, Shamir, Adleman).
- Choose 2 large prime numbers $p$ and $q$.
- $n = p \times q$ should be about 1024 bits long.
- $\phi(n) = (p-1)(q-1)$.
- Choose $e$ such that $e \phi(n)$ has no common factors with $n$.
- Find $d$ such that $(d \times e) \mod \phi(n) = 1$.
- Public key is $(n, e)$, private key is $(n, d)$.
- Message $m$ is encrypted to $c = m^e \mod n$.
- Ciphertext $c$ is decrypted $m = c^d \mod n$.

RSA example:

A host chooses $p=5$, $q=7$. Then $n=35$, $\phi(n)=24$.
- $e=5$ (so $e$, $\phi(n)$ relatively prime).
- $d=29$ (so $ed \equiv 1 \mod \phi(n)$).

<table>
<thead>
<tr>
<th>Encrypt</th>
<th>Decrypt</th>
</tr>
</thead>
<tbody>
<tr>
<td>letter</td>
<td>$m$</td>
</tr>
<tr>
<td>m</td>
<td>12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Encrypt</th>
<th>Decrypt</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c$</td>
<td>$c^d \mod n$</td>
</tr>
<tr>
<td>17</td>
<td>$451145602194734359150941150220230808$</td>
</tr>
</tbody>
</table>

Public-Key Cryptography (5)

- A 512-bit number (155 decimals) was factored into two primes in 1999 using one Cray and 300 workstations.
- 1024-bit keys are still safe.
- Incredibly useful property of public-key cryptography:
  - $m^d \mod n = (m^e)^d \mod n = (m^d)^e \mod n$.
  - Thus, can swap the order in which the keys are used.
- Example: can use the private key for encryption and a public key for decryption – will see how it is useful in authentication!