CSE597B: Special Topics in Network and Systems Security

Introduction to Modern Cryptography (1)

Instructor: Sencun Zhu

Outline

• Basic concepts and terminology
• Security objectives
• Overview of symmetric key cryptography
• Overview of public key cryptography
• Security attacks
• Security strength
• Properties and principles
• Resources

Basic Concepts

• Cryptography
  – The study of mathematical techniques related to aspects of information security
  – About communication in the presence of adversaries
• Cryptanalysis
  – The study of methods for obtaining the meaning of encrypted information without accessing the secret information
• Cryptology
  – Cryptography + cryptanalysis
• Cipher
  – An algorithm for performing encryption/decryption
Terminology

- **Cryptographic primitive**
  - Cryptographic tools
    - E.g., encryption schemes, signatures, hash functions
- **Information security service**
  - A method for providing a certain security objective
    - E.g., confidentiality, integrity, availability
- **Cryptographic protocol**
  - A distributed algorithm defined by a sequence of steps specifying the actions of two or more entities
    - E.g., Kerberos, SSL, IPSEC
- **Security mechanism**
  - Including protocols, algorithms, and non-cryptographic techniques

Security Objectives

- **Confidentiality**
  - A service keeping information from all but those authorized to have it
    - Synonym: secrecy, privacy
- **Authentication**
  - A service determining whether someone or something is, in fact, who or what it is declared to be
    - Including entity authentication and data origin (source) authentication
- **Data integrity**
  - A service determining whether data is altered by unauthorized parties (implicitly provided by data source authentication)
- **Availability**
  - A service providing timely accessibility of data to authorized parties; otherwise, called denial of service (DoS)
- **Non-repudiation**
  - A service preventing an entity from denying previous commitments or actions
- **Access control**
  - A service determining who and what an entity can access
- **Anonymity**
  - A service protecting the identity of an entity from others
Encryption/Decryption

- **Plaintext**: a message in its original form
- **Ciphertext**: a message in the transformed, unrecognized form
- **Encryption**: the process that transforms a plaintext into a ciphertext
- **Decryption**: the process that transforms a ciphertext to the corresponding plaintext
- **Key**: the secret value used to control encryption/decryption

**Symmetric Key Encryption**

- **Same key** is used for encryption and decryption
- **Also known as**
  - private encryption
  - Conventional encryption
- **Including block cipher and stream cipher**

**Requirements**

- Two requirements for secure use of symmetric encryption:
  - a strong encryption algorithm
  - a secret key known only to sender / receiver
  - \( Y = E_K(X) \)
  - \( X = D_K(Y) \)
- Assume encryption algorithm is publicly known
- Implies a secure channel to distribute key
**Block Cipher**

- **Encrypt**
  - Plaintext block of length \( N \)
  - Secret key
  - Cipher block of length \( N \)
- **Decrypt**

- \( N: 64, 128, 256 \) bits, for example
- **Bijection**

**Block Cipher (2)**

- **Classes of block ciphers**
  - Substitution cipher
    - Replace symbols
  - Transposition (permutation) cipher
    - Rearrange the order of the symbols
  - Product cipher: the combination of the other two
- **Round**
  - The composition of a substitution and a transposition
- Most modern block cipher systems apply a number of rounds in succession
  - E.g., DES has 16 rounds
- **Examples**
  - DES, 3DES, AES, Serpent, Twofish

**Stream Cipher**

- **key**
- **Pseudo random number generator**
  - Bitwise \( \oplus \)
- **plaintext**
  - 101011101101110011...
- **ciphertext**
  - 001010010100110011...

- Encrypt messages bit-by-bit
  - No need buffering, no error propagation
- **Examples**
  - RC4, one-time pad
One-time Pad

- A stream cipher, also called Vernam Cipher
- Definition
  - A binary plaintext string: \(m_1 \ m_2 \ m_3 \ldots m_t\)
  - A binary key string: \(k_1 \ k_2 \ k_3 \ldots k_t\)
  - The ciphertext string \(c_1 \ c_2 \ c_3 \ldots c_t\), where
    \[c_i = m_i \text{ xor } k_i\]
  - The key string is randomly chosen and never used again
- Proved to be theoretically unbreakable
  - An unbreakable system requires a random key of the same length as the message

Message Authentication Code

- MAC
  - Also known as cryptographic checksum, Message Integrity Code (MIC)
  - Assume that the sender and the receiver share a common secret key
  - A small fixed-size block, called authentication tag, generated from the message with secret key cryptography
  - The tag is appended to the original message

Message Authentication Code (2)

- Requirement
  - For \(M\) and \(F_K(M)\), it’s computationally infeasible to construct a message \(M'\) such that \(F_K(M') = F_K(M)\).
- Examples
  - DES CBC mode
  - HMAC, based on secure hash function
Public-Key Encryption

- Public key is known by everybody
  - Everybody can use it to send a message to Alice securely
- Private key is only known by the key owner Alice
  - Only Alice can decrypt the message

Digital Signature

- Only the party with the private key can create a digital signature
- The digital signature is verifiable by anyone who knows the public key.
- The signer cannot deny that he/she has done so.

Public-Key Cryptography

- Public-key/two-key/asymmetric cryptography involves the use of two keys:
  - A public-key, which may be known by anybody, and can be used to encrypt messages, and verify signatures
  - A private-key, known only to the recipient, used to decrypt messages, and sign (create) signatures
- Asymmetric
  - Those who encrypt messages or verify signatures cannot decrypt messages or create signatures
Public-Key Cryptography

- Why Public-Key Cryptography?
  - key distribution – how to have secure communications in general without having to trust a KDC with your key
  - digital signatures – how to verify a message comes intact from the claimed sender

- Applications
  - encryption/decryption (provide secrecy)
  - digital signatures (provide authentication)
  - key exchange (of session keys)

- Example algorithms
  - RSA, DSA, Diffie-Hellman, Elliptic curve

Classify Security Attacks

- Passive attacks
  - Obtain message contents or monitor traffic flows
  - Only threaten confidentiality of data

- Active attacks
  - Masquerade of one entity as some other
  - Replay previous messages
  - Modify messages in transit
  - Add, delete messages
  - Denial of service

Brute Force Search

- Always possible to simply try every key
- Most basic attack, proportional to key size
- Assume either know / recognise plaintext

<table>
<thead>
<tr>
<th>Key Length (bits)</th>
<th>Software Attack</th>
<th>Hardware Attack</th>
<th>Intelligence Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>1 day</td>
<td>1 minute</td>
<td>1 second</td>
</tr>
<tr>
<td>56</td>
<td>13 years</td>
<td>2 months</td>
<td>1 day</td>
</tr>
<tr>
<td>64</td>
<td>26 years</td>
<td>10 years</td>
<td>7 months</td>
</tr>
<tr>
<td>128</td>
<td>130 years</td>
<td>115 years</td>
<td>105 years</td>
</tr>
</tbody>
</table>
Cryptanalysis

- Ciphertext only:
  - Analyze only with the ciphertext
  - The most difficult type of attack
- Known plaintext:
  - Secret may be revealed (by spy, time, pattern), thus <ciphertext, plaintext> pair is obtained
- chosen plaintext
  - Select plaintext and obtain ciphertext to attack cipher
- chosen ciphertext
  - Select ciphertext and obtain plaintext to attack cipher

Birthday Attacks

- Birthday paradox
  - if there are 23 people in a room then there is a slightly more than 50% chance that at least two of them will have the same birthday
- Collision
  - $x_1 \neq x_2$, but $f(x_1) = f(x_2)$
- Birthday attack
  - Wait for a single value occur twice (i.e., a collision) within the same set of elements
  - Need approximately $\sqrt[2]{N}$ observations to see a collision with 50% chance, if the size of a set is $N$
  - Replay messages

Meet-in-the-Middle Attacks

- Actively build a table of keys by yourself
- Compute $f(k,m)$ for every key $k$
  - $f$ is an encryption or authentication function
  - $m$ is a known message
- eavesdrop a value $f(k',m)$
- If $f(k',m) = f(k, m)$, i.e., a collision, there is a good chance $K = k$
- Thus an attacker can insert or decrypt any messages
  - In the birthday attack, it can only replay messages
- The probability of a collision is the same as in birthday attack
- See an example below
An Example Attack

- **DES**
  - A well known block cipher used for many years
  - Key size 56 bits, block size 64 bits
  - not secure any longer
- **Double DES**
  - Encrypt the plaintext twice with two different DES keys
  - Key length increases to 112 bits
- **Two concerns**
  - Is DES a group?
    - \( E_{k_2}(E_{k_1}(P)) = E_{k_3}(P) \)
    - Implication?
  - Subject to meet-in-the-middle attack

Attack on Double DES

- For a known pair (P, C)
  - Encrypt P for all 2^{56} values for \( K_1 \)
  - Decrypt C for all 2^{56} values for \( K_2 \), and for each result check the table
  - A match reveals a possible combination of key

\[ X = E_{k_1}(P) = D_{k_2}(C) \]

Attack Analysis

- **Analysis**
  - With one pair (P, C), #keys that can survive the test is \( 2^{112}/2^{64} = 2^{48} \)
  - With another pairs (P', C'), #keys that can survive both tests is \( 2^{48}/2^{64} = 2^{-16} \)
  - The probability that the correct keys are determined is 1 - 2^{-16}
- **Goal of double DES**
  - Increase the difficulty of exhaustive key search (2^{112} keys)
  - In effect, the difficulty is on the order of 2^{56}
- **Solution:** using Triple DES
Security Definitions

• Unconditional security
  – No matter how much computer power is available, an adversary cannot defeat the system
  – Perfect secrecy
    • Unconditional security for encryption schemes
    • Observation of ciphertext provides no information to an adversary, i.e., entropy is not reduced
    • Symmetric key encryption
      – The key must be at least the same size as the message (one-time pad)
      – Practical encryption schemes do not provide perfect secrecy
    • Public key encryption
      – Do not provide perfect secrecy

Security Definitions (2)

• Computational security
  – The perceived level of computation required to defeat a security system using the best known attack exceeds, by a comfortable margin, the computational resources of the hypothesized adversary
    – e.g., given limited computing resources, it takes the age of universe to break a cipher
  • Provable security
    – The difficulty of defeating a system can be shown to be essentially as difficult as solving a well-known and supposedly difficult problem (e.g., integer factorisation or discrete logarithms)

Security Definitions (3)

• Ad hoc Security
  – Most common for protocols
  – Claims of security generally remain questionable and unforeseen attacks remain a threat
Properties and Principles

• The weakest link property
  – A security system is only as strong as its weakest link

• Kerchoffs’ Principle
  – The security of an encryption scheme must depend on the secrecy of the key, and not on the secrecy of the algorithm

• Encryption does not provide authentication
  – Nonsense could do harm

• Encryption should be combined with authentication
  – Encryption-first or authentication-first?

• Do not use the same key for both encryption and authentication
  – From a secret deriving two keys (e.g., based on a pseudo random function) for encryption and authentication, respectively

Resources

• Wikipedia
  – http://en.wikipedia.org/wiki/Main_Page
  – Very good for concept lookup

• Citeseer
  – http://citeseer.ist.psu.edu
  – Good for search and download papers
  – Find the citations of papers and related papers

• Google
  – Good for finding everything

• ACM and IEEE digital libraries