Chapter 2
Cryptographic Tools

Symmetric Encryption

- The universal technique for providing confidentiality for transmitted or stored data
- Also referred to as conventional encryption or single-key encryption
- Two requirements for secure use:
  - Need a strong encryption algorithm
  - Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure
Attacking Symmetric Encryption

Cryptanalytic Attacks
- Rely on:
  - Nature of the algorithm
  - Some knowledge of the general characteristics of the plaintext
  - Some sample plaintext-ciphertext pairs
- Exploit the characteristics of the algorithm to attempt to deduce a specific plaintext or the key being used
- If successful all future and past messages encrypted with that key are compromised

Brute-Force Attack
- Try all possible keys on some ciphertext until an intelligible translation into plaintext is obtained
- On average half of all possible keys must be tried to achieve success

Table 2.1

<table>
<thead>
<tr>
<th></th>
<th>DES</th>
<th>Triple DES</th>
<th>AES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plaintext block size (bits)</td>
<td>64</td>
<td>64</td>
<td>128</td>
</tr>
<tr>
<td>Ciphertext block size (bits)</td>
<td>64</td>
<td>64</td>
<td>128</td>
</tr>
<tr>
<td>Key size (bits)</td>
<td>56</td>
<td>112 or 168</td>
<td>128, 192, or 256</td>
</tr>
</tbody>
</table>

DES = Data Encryption Standard
AES = Advanced Encryption Standard

Comparison of Three Popular Symmetric Encryption Algorithms
Data Encryption Standard (DES)

- The most widely used encryption scheme
- FIPS PUB 46
- Referred to as the Data Encryption Algorithm (DEA)
- Uses 64 bit plaintext block and 56 bit key to produce a 64 bit ciphertext block

Strength concerns:
- Concerns about algorithm
- DES is the most studied encryption algorithm in existence
- Use of 56-bit key
- Electronic Frontier Foundation (EFF) announced in July 1998 that it had broken a DES encryption

Table 2.2

<table>
<thead>
<tr>
<th>Key size (bits)</th>
<th>Cipher</th>
<th>Number of Alternative Keys</th>
<th>Time Required at 10^9 Decryptions/s</th>
<th>Time Required at 10^9 Decryptions/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>DES</td>
<td>$2^{56}$ = 7.2 x 10^16</td>
<td>$2^{56}$ x 1.125 years</td>
<td>1 hour</td>
</tr>
<tr>
<td>128</td>
<td>AES</td>
<td>$2^{128}$ = 3.4 x 10^38</td>
<td>$2^{128}$ m = 5.3 x 10^23 years</td>
<td>5.3 x 10^17 years</td>
</tr>
<tr>
<td>168</td>
<td>Triple DES</td>
<td>$2^{168}$ = 3.7 x 10^48</td>
<td>$2^{168}$ m = 5.8 x 10^23 years</td>
<td>5.8 x 10^18 years</td>
</tr>
<tr>
<td>192</td>
<td>AES</td>
<td>$2^{192}$ = 6.3 x 10^57</td>
<td>$2^{192}$ m = 9.8 x 10^26 years</td>
<td>9.8 x 10^19 years</td>
</tr>
<tr>
<td>256</td>
<td>AES</td>
<td>$2^{256}$ = 1.2 x 10^77</td>
<td>$2^{256}$ m = 1.8 x 10^49 years</td>
<td>1.8 x 10^20 years</td>
</tr>
</tbody>
</table>

Average Time Required for Exhaustive Key Search

Triple DES (3DES)

- Repeats basic DES algorithm three times using either two or three unique keys
- First standardized for use in financial applications in ANSI standard X9.17 in 1985
- Attractions:
  - 168-bit key length overcomes the vulnerability to brute-force attack of DES
  - Underlying encryption algorithm is the same as in DES
- Drawbacks:
  - Algorithm is sluggish in software
  - Uses a 64-bit block size
Advanced Encryption
Standard (AES)

Needed a
replacement for
3DES

NIST called for
proposals for a
new AES in 1997

Selected
Rijndael in
November 2001

3DES was not
reasonable for
long term use

Should have a security
strength equal to or
better than 3DES

Published as
FIPS 197

Should have significantly
improved efficiency

Symmetric block cipher

128-bit data and
128/192/256-bit keys

Practical Security Issues

- Typically symmetric encryption is applied to a unit of
data larger than a single 64-bit or 128-bit block
- Electronic codebook (ECB) mode is the simplest
approach to multiple-block encryption
  - Each block of plaintext is encrypted using the same key
  - Cryptanalysts may be able to exploit regularities in the
plaintext
- Modes of operation
  - Alternative techniques developed to increase the security
of symmetric block encryption for large sequences
  - Overcomes the weaknesses of ECB

Figure 2.2: Types of Symmetric Encryption
Block & Stream Ciphers

**Block Cipher**
- Processes the input one block of elements at a time
- Produces an output block for each input block
- Can reuse keys
- More common

**Stream Cipher**
- Processes the input elements continuously
- Produces output one element at a time
- Primary advantage is that they are almost always faster and use far less code
- Encrypts plaintext one byte at a time
- Pseudorandom stream is one that is unpredictable without knowledge of the input key

Message Authentication

- Protects against active attacks
- Verifies received message is authentic
- Can use conventional encryption
- Contents have not been altered
- From authentic source
- Timely and in correct sequence
- Only sender & receiver share a key

Figure 2.3 Message Authentication Using a Message Authentication Code (MAC).
Hash Function Requirements

- Can be applied to a block of data of any size
- Produces a fixed-length output
- \( H(x) \) is relatively easy to compute for any given \( x \)
- One-way or pre-image resistant
  - Computationally infeasible to find any \( x \) such that \( H(x) = h \)
- Computationally infeasible to find \( y \) such that \( H(y) = H(x) \)
- Collision resistant or strong collision resistance
  - Computationally infeasible to find any \( x \) and \( y \) such that \( H(x) = H(y) \)
Security of Hash Functions

There are two approaches to attacking a secure hash function:

- Cryptanalysis
  - Exploit logical weaknesses in the algorithm
- Brute-force attack
  - Through chosen function with a hash output that was produced by the algorithm

Additional secure hash function applications:

- SHA: Most widely used hash algorithm
- Passwords
  - Hash of a password is stored by an operating system
- Intrusion detection
  - Store H(F) for each file on a system and secure the hash values

Public-Key Encryption Structure

- Publicly proposed by Diffie and Hellman in 1976
- Asymmetric
  - Uses two separate keys
    - Public key and private key
    - Public key is made public for others to use
  - Based on mathematical functions

Some form of protocol is needed for distribution

- Plaintext
  - Readable message or data that is fed into the algorithm as input
- Encryption algorithm
  - Performs transformations on the plaintext
- Public and private key pair
  - One key for encryption, one for decryption
- Ciphertext
  - Scrambled message produced as output
- Decryption key
  - Produces the original plaintext
User encrypts data using his or her own private key.

Anyone who knows the corresponding public key will be able to decrypt the message.

Table 2.3

Applications for Public-Key Cryptosystems

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Digital Signature</th>
<th>Symmetric Key Distribution</th>
<th>Encryption of Secret Keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Diffie-Hellman</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DSS</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Elliptic Curve</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Requirements for Public-Key Cryptosystems

- Computationally easy to create key pairs
- Useful if either key can be used for each role
- Computationally infeasible for opponent to otherwise recover digital message
- Computationally easy for sender knowing public key to encrypt messages
- Computationally easy for receiver knowing private key to decrypt ciphertext
- Computationally infeasible for opponent to determine private key from public key
Digital Signatures

- Used for authenticating both source and data integrity
- Created by encrypting hash code with private key
- Does not provide confidentiality
  - Even in the case of complete encryption
  - Message is safe from alteration but not eavesdropping

Figure 2.7 Public-Key Certificate Use
Digital Envelopes

- Protects a message without needing to first arrange for sender and receiver to have the same secret key.
- Equates to the same thing as a sealed envelope containing an unsigned letter.

Random Numbers

- Keys for public-key algorithms
- Stream key for symmetric stream cipher
- Symmetric key for use as a temporary session key or in creating a digital envelope
- Handshaking to prevent replay attacks
- Session key

Random Number Requirements

- Each number is statistically independent of other numbers in the sequence.
- Each number is not predictable in the future.
- No one value in the sequence can be inferred from the others.
- Opponent should not be able to predict future elements of the sequence on the basis of earlier elements.
Random versus Pseudorandom

Cryptographic applications typically make use of algorithmic techniques for random number generation.

- Algorithmic techniques produce sequences of numbers that are not statistically random.

Random numbers are:

- Used in algorithmic sources to produce randomness.
- Often adequate for many applications.
- Easily predictable.

Pseudorandom numbers are:

- Sequences produced that satisfy statistical randomness tests.
- Likely to be predictable.

True random number generator (TRNG): uses a nondeterministic source to produce randomness.

- TRNGs use unpredictable natural processes.
- Increasingly provided on modern processors.

Practical Application: Encryption of Stored Data

Common to encrypt transmitted data:

- Digital signatures
- Public-key certificates
- Secure hash functions

Much less common for stored data:

- Database encryption
- Application layer encryption
- Symmetric encryption
- High performance

Approaches to encrypt stored data:

- Use a commercially available encryption library.
- Back-end appliance.
- Library-based tape encryption.
- Background laptop/PC data encryption.

Summary

- Confidentiality with symmetric encryption.
  - Symmetric encryption
  - Symmetric block encryption
  - Stream ciphers
- Message authentication and hash functions.
  - Authentication using symmetric encryption
  - Message authentication without message encryption
  - Secure hash functions
- Other applications of hash functions.
- Random and pseudorandom numbers.
  - The use of random numbers
  - Random versus pseudorandom numbers