ECE297:11 - Lecture 2

Types of Cryptosystems

Implementation of Security Services

Secret-key vs. public-key ciphers
Secret-key (Symmetric) Cryptosystems

key of Alice and Bob - $K_{AB}$

Network

Alice

Encryption

Bob

Decryption

Key Distribution Problem

User 1

User 2

User 3

User 4

User 5

$N - Users \rightarrow \frac{N \cdot (N-1)}{2} \text{ Keys}$

<table>
<thead>
<tr>
<th>Users</th>
<th>Keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>5,000</td>
</tr>
<tr>
<td>1000</td>
<td>500,000</td>
</tr>
</tbody>
</table>
Digital Signature Problem

Both corresponding sides have the same information and are able to generate a signature

There is a possibility of the
- receiver falsifying the message
- sender denying that he/she sent the message

Public Key (Asymmetric) Cryptosystems

Public key of Bob - $K_B$
Private key of Bob - $K_B$

Alice
Network
Decryption
Bob
Encryption

Public Key
- Asymmetric Cryptosystem
- Public key - $K_B$
- Private key - $K_B$
### Classification of cryptosystems

#### Terminology

- **secret-key**
- **public key**
- **symmetric**
- **asymmetric**
- **symmetric-key**
- **classical**
- **conventional**

### One-way function

**Example:**

\[
f: Y = f(X) = A^X \mod P
\]

where $P$ and $A$ are constants, $P$ is a large prime, and $A$ is an integer smaller than $P$

<table>
<thead>
<tr>
<th>Number of bits of $P$</th>
<th>Average number of multiplications necessary to compute $f$</th>
<th>$f^{-1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>1500</td>
<td>$10^{30}$</td>
</tr>
</tbody>
</table>
Trap-door one-way function

Whitfield Diffie and Martin Hellman
“New directions in cryptography,” 1976

PUBLIC KEY

X → f(X) → Y

PRIVATE KEY

f⁻¹(Y)

Key Distribution

Alice
message

Bob
message

Bob’s public key

Bob’s private key

Intruder

message

Bob’s public key

ciphertext

ciphertext

ciphertext
Digital Signature

Alice

signature

Alice’s private key

message

Alice’s public key

signatures

message

Alice’s public key

Intruder

signature

Alice’s public key

message

Alice’s private key

Judge

signature

Alice’s public key

message

Implementation of Security Services
Hash function

 arbitary length

 message

 hash function

 hash value

 fixed length
Hash functions

Basic requirements

1. Public description, NO key
2. Compression
   arbitrary length input $\rightarrow$ fixed length output
3. Ease of computation

Hash functions

Security requirements

<table>
<thead>
<tr>
<th>It is computationally infeasible</th>
<th>Given</th>
<th>To Find</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Preimage resistance</td>
<td>$h(m)$</td>
<td>$m$</td>
</tr>
<tr>
<td>2. 2nd preimage resistance</td>
<td>$m$ and $h(m)$</td>
<td>$m' \neq m$, such that $h(m') = h(m)$</td>
</tr>
<tr>
<td>3. Collision resistance</td>
<td></td>
<td>$m' \neq m$, such that $h(m') = h(m)$</td>
</tr>
</tbody>
</table>
Brute force attack against One-Way Hash Function

Given $y$

\[ m_i' \]

\[ h \]

\[ h(m_i') = y \]

$i=1..2^n$

$2^n$ messages with the contents required by the forger

Creating multiple versions of the required message

I \{state confirm\} \{thereby\} that I \{borrowed\} received

\{\$10,000 \{ten thousand dollars\}\} from \{Mr. Dr. Kris\} Krzysztof

Gaj on \{June 4, \06/04\} \{2002\}. This \{money sum of money\}

\{should is required to\} be \{returned given back\} to \{Mr. Dr.\} Gaj

by the \{end middle\} of \{June July\}. 
**Brute force attack against**

**Collision Resistant Hash Function**

$r$ messages acceptable for the signer

$m_i$ \( i = 1 .. r \)

$h(m_i) \leftarrow n \text{ - bits}$

$h(m_i) = h(m'_i)$

$r$ messages required by the forger

$m'_j$ \( j = 1 .. r \)

$h(m'_j) \leftarrow n \text{ - bits}$

**Message required by the forger**

I state thereby that I borrowed received

\{ $10,000$ \} from Mr. Kris Krzysztof

\{ ten thousand dollars \} Dr. 

Gaj on June 4, 06 / 04 2002. This money sum of money

\{ should be returned given back \} to Mr. Dr. Gaj

\{ is required to \} by the end middle of July June .
Message acceptable for the signer

I state thereby that on June 4, 2001.

I borrowed received from Mr. Dr. Kris Krzysztof a book manuscript.

on fast efficient implementations of ciphers cryptosystems.

This text book should be returned given back.

to Mr. Dr. Gaj by the end of November.

Birthday paradox

How many students there must be in a class for there be a greater than 50% chance that

1. one of the students shares the teacher’s birthday (day and month)?

2. any two of the students share the same birthday (day and month)?
Birthday paradox

How many students there must be in a class for there be a greater than 50% chance that

1. one of the students shares the teacher’s birthday (day and month)?

\[ \sim \frac{366}{2} = 188 \]

2. any two of the students share the same birthday (day and month)?

\[ \sim \sqrt{366} \approx 19 \]

Authentication

Alice

- Message
- MAC
- Secret key algorithm
- Secret key of Alice and Bob

Bob

- Message
- MAC
- Secret key algorithm
- Secret key of Alice and Bob
- MAC’
- yes
- no
MAC - Message Authentication Codes
(keyed hash functions)

**Basic requirements**

1. Public description, SECRET key parameter
2. Compression
   
   arbitrary length input $\rightarrow$ fixed length output
3. Ease of computation
MAC functions

Security requirements

Given zero or more pairs

\[ m_i, \text{MAC}(m_i) \quad i = 1..k \]

it is computationally impossible to find any new pair

\[ m’, \text{MAC}(m’) \]

Such that

\[ m’ \neq m_i \quad i = 1..k \]

---

CBC-MAC (Cipher Block Chaining MAC)
Relations among security services

- **INTEGRITY**
- **AUTHENTICATION**
- **NON-REPUDIATION**
- **CONFIDENTIALITY**

**Message Hash Function**

Alice
- Message
- Signature
- Hash function
- Hash value
- Public key algorithm
- Alice’s private key

Bob
- Message
- Signature
- Hash function
- Hash value 1
- Yes
- Hash value 2
- No
- Public key algorithm
- Alice’s public key
Authentication

Alice

Message → MAC

$K_{AB}$ → Secret key algorithm

Secret key of Alice and Bob

Bob

Message → MAC

$K_{AB}$ → Secret key algorithm

Secret key of Alice and Bob

MAC’ → yes

MAC → no

Hybrid Systems
Features required from today’s ciphers

STRENGTH → PERFORMANCE

FUNCTIONALITY
- easy key distribution
- digital signatures

Features of secret-key ciphers

STRENGTH → PERFORMANCE

FUNCTIONALITY
- easy key distribution
- digital signatures
Features of public-key ciphers

- **STRENGTH**
- **FUNCTIONALITY**
  - easy key distribution
  - digital signatures
- **PERFORMANCE**

Ciphering and Deciphering Speed

*on average for implementations based on the same technology*

<table>
<thead>
<tr>
<th>Ciphering Speed</th>
<th>Software</th>
<th>Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>DES deciphering speed</td>
<td>≈ 100</td>
<td></td>
</tr>
<tr>
<td>RSA deciphering speed</td>
<td></td>
<td>≈ 1000</td>
</tr>
</tbody>
</table>
### Basic operations of secret key ciphers - S-box

**S-box n x m**

**Software**

\[
\text{C}
\]

\[
\text{WORD } S[1\leq n]=
\{ 0x23, 0x34, 0x56
\ldots
\}
\]

**ASM**

\[
\text{S DW } 23H, 34H, 56H
\]

.....

**Hardware**

- **ROM**
  - \( n \)-bit address
  - 2\(^n \) words
  - \( m \)-bit output

- **direct logic**

\[
\begin{align*}
&x_1 \quad x_2 \quad \ldots \quad x_n \\
&\vdots \\
&y_1 \quad y_2 \quad \ldots \quad y_m
\end{align*}
\]

### Basic operations of secret key ciphers - P-box

**P-box n x n**

**Software**

\[
\text{C}
\]

\[
\text{sequence of instructions}
\]

\[
\lll, |, &
\]

**ASM**

\[
\begin{align*}
&x_1 \quad x_2 \quad x_3 \quad x_{n-1} \quad x_n \\
&\vdots \\
&y_1 \quad y_2 \quad y_3 \quad y_{n-1} \quad y_n
\end{align*}
\]

**Hardware**

- **sequence of instructions**
  - ROL, OR, AND

- **order of wires**
RSA as a trap-door one-way function

message \( m \)

\[ c = f(m) = m^e \mod n \]

\[ m = f^{-1}(c) = c^d \mod n \]

ciphertext \( c \)

PUBLIC KEY

PRIVATE KEY

\( n = p \cdot q \)
\( p, q - \text{large prime numbers} \)
\( e \cdot d \equiv 1 \mod ((p-1)(q-1)) \)

RSA keys

PUBLIC KEY

\{ e, n \}

PRIVATE KEY

\{ d, p, q \}

\( n = p \cdot q \)
\( p, q - \text{large prime numbers} \)
\( e \cdot d \equiv 1 \mod ((p-1)(q-1)) \)
Basic Operations of the Public Key Cryptosystem RSA

Encryption

\[ \text{ciphertext} = \text{plaintext} \mod \text{public key modulus} \]

Decryption

\[ \text{ciphertext} = \text{plaintext} \mod \text{private key modulus} \]

Hybrid Systems

Alice

Bob

Bob’s public key  session key  random
Encrypted session key  Encrypted message
RSA public key  DES session key

Encrypted session key  Encrypted message

Bob’s private key
Hybrid Systems - Sender’s Side (2)

Alice

message

<table>
<thead>
<tr>
<th>session key</th>
<th>random</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public key cipher</td>
<td></td>
</tr>
</tbody>
</table>

Bob’s public key

| Encrypted session key | Encrypted message |

Hybrid Systems - Receiver’s Side (2)

Bob

message

<table>
<thead>
<tr>
<th>session key</th>
<th>random</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public key cipher</td>
<td></td>
</tr>
</tbody>
</table>

Bob’s private key

| Encrypted session key | Encrypted message |
Block vs. stream ciphers

Every block of ciphertext is a function of only one corresponding block of plaintext.

Every block of ciphertext is a function of the current and all proceeding blocks of plaintext.
Typical stream cipher

Sender

key initialization vector

Pseudorandom Key Generator

\( k_i \) keystream

\( m_i \) plaintext \( c_i \) ciphertext

Receiver

key initialization vector

Pseudorandom Key Generator

\( k_i \) keystream

\( c_i \) ciphertext \( m_i \) plaintext

Evaluating the security of secret-key ciphers
Classification of attacks (1)

**Ciphertext-only attack**

*Given:*  
ciphertext

*Looked for:*  
plaintext  
or key

*Example:*  
Frequency analysis of letters in the ciphertext  
(effective only for most simple historical ciphers)

Classification of attacks (2)

**Known plaintext attack**

*Given:*  
ciphertext  
guessed fragment of the plaintext

*Looked for:*  
remaining plaintext  
or key

*Example:*  
 exhaustive key search  
(brute-force) attack  
successive keys  
cipher
Classification of attacks (3)

**Chosen plaintext attack**

*Given:*
- Capability to encipher an arbitrarily chosen fragment of the plaintext

*Looked for:*
- Key

*Example:*

Differential cryptanalysis

![Diagram of chosen plaintext attack with equations](image)

Classification of attacks (4)

**Chosen ciphertext attack**

*Given:*
- Capability to decipher an arbitrarily chosen fragment of the ciphertext

*Looked for:*
- Key

![Diagram of chosen ciphertext attack with equations](image)