Basic Cryptography

Network Security
Objectives

- Understand/explain/employ the basic cryptographic techniques
  - Review the basic number theory used in cryptosystems
  - Classical system
  - Public-key system
  - Some crypto analysis
  - Message digest
Secure Information Transmission
(network security model)

Sender

Message

Secret Information

Secure Message

Information channel

Opponent

Receiver

Message

Secret Information

Secure Message

Trusted Third Party
arbiter, distributor of
secret information

Security related
transformation
Security of Information Systems
(Network access model)

- Opponent
  - hackers
  - software

- Access Channel

- Gatekeeper
  - firewall or equivalent, password-based login

- Internal Security Control
  - Access control, Logs, audits, virus scans etc.
Issues in Network security

- Distribution of secret information to enable secure exchange of information
- Effect of communication protocols needs to be considered
- Encryption *if used cleverly and correctly*, can provide several of the security services
- Physical and logical placement of security mechanisms
- Countermeasures need to be considered
Cryptology

CRYPTOLOGY

CRYPTOGRAPHY

Private Key (Secret Key)
- Block Cipher
- Stream Cipher

Public Key
- Encipher, encrypt
- Decipher, decrypt
- Integer Factorization
- Discrete Logarithm

CRYPTANALYSIS
Elementary Number Theory

- Natural numbers $\mathbb{N} = \{1, 2, 3, \ldots\}$
- Whole numbers $\mathbb{W} = \{0, 1, 2, 3, \ldots\}$
- Integers $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, 3, \ldots\}$
- Divisors
  - A number $b$ is said to divide $a$ if $a = mb$ for some $m$ where $a, b, m \in \mathbb{Z}$
  - We write this as $b \mid a$
    - Read as “$b$ divides $a$”
Divisors

- Some common properties
  - If $a \mid 1$, $a = +1$ or $-1$
  - If $a \mid b$ and $b \mid a$ then $a = +b$ or $-b$
  - Any $b \in \mathbb{Z}$ divides 0 if $b \neq 0$
  - If $b \mid g$ and $b \mid h$ then $b \mid (mg + nh)$ where $b, m, n, g, h \in \mathbb{Z}$

- Examples:
  - The positive divisors of 42 are 1, 2, 3, 6, 7, 14, 21, 42
  - 3|6 and 3|21 => 3|21m+6n for $m, n \in \mathbb{Z}$
Prime Numbers

- An integer \( p \) is said to be a prime number if its only positive divisors are 1 and itself
  - 2, 3, 7, 11, ..
- Any integer can be expressed as a \textit{unique} product of prime numbers raised to positive integral powers
- Examples
  - \( 7569 = 3 \times 3 \times 29 \times 29 = 3^2 \times 29^2 \)
  - \( 5886 = 2 \times 27 \times 109 = 2 \times 3^3 \times 109 \)
  - \( 4900 = 7^2 \times 5^2 \times 2^2 \)
  - \( 100 = ? \)
  - \( 250 = ? \)
- This process is called \textit{Prime Factorization}
Definition: Greatest Common Divisor
- This is the largest divisor of both \(a\) and \(b\).

Given two integers \(a\) and \(b\), the positive integer \(c\) is called their GCD or greatest common divisor if and only if
  - \(c \mid a\) and \(c \mid b\)
  - Any divisor of both \(a\) and \(b\) also divides \(c\)

Notation: \(\text{gcd}(a, b) = c\)

Example: \(\text{gcd}(49, 63) = ?\)
Relatively Prime Numbers

- Two numbers are said to be relatively prime if their gcd is 1
  - Example: 63 and 22 are relatively prime
- How do you determine if two numbers are relatively prime?
  - Find their GCD or
  - Find their prime factors
    - If they do not have a common prime factor other than 1, they are relatively prime
  - Example: $63 = 9 \times 7 = 3^2 \times 7$ and $22 = 11 \times 2$
The modulo operation

- What is 27 mod 5?

Definition

- Let $a, r, m$ be integers and let $m > 0$
- We write $a \equiv r \mod m$ if $m$ divides $r - a$ (or $a - r$) and $0 \leq r < m$
- $m$ is called the modulus
- $r$ is called the remainder
- Note that $a = m.q + r$ where $q$ is another integer (quotient)
Modular Arithmetic

- We say that $a \equiv b \mod m$ if $m \mid a - b$
  - Read as: $a$ is congruent to $b$ modulo $m$
  - $m$ is called the modulus
  - Example: $27 \equiv 2 \mod 5$

- Note that $b$ is the *remainder* after dividing $a$ by $m$
  - Example: $27 \equiv 7 \mod 5$ and $7 \equiv 2 \mod 5$

- $a \equiv b \mod m \implies b \equiv a \mod m$
  - Example: $2 \equiv 27 \mod 5$

- We usually consider the *smallest positive remainder* which is sometimes called the *residue*
Modulo Operation

- The modulo operation “reduces” the infinite set of integers to a finite set
- Example: modulo 5 operation
  - We have five sets
    - \{..., -10, -5, 0, 5, 10, ...\} => \(a \equiv 0 \mod 5\)
    - \{..., -9, -4, 1, 6, 11, ...\} => \(a \equiv 1 \mod 5\)
    - \{..., -8, -3, 2, 7, 12, ...\} => \(a \equiv 2 \mod 5\), etc.
  - The set of residues of integers modulo 5 has five elements \{0, 1, 2, 3, 4\} and is denoted \(\mathbb{Z}_5\).
Modulo Operation

- Properties
  - \[ [(a \mod n) + (b \mod n)] \mod n = (a + b) \mod n \]
  - \[ [(a \mod n) - (b \mod n)] \mod n = (a - b) \mod n \]
  - \[ [(a \mod n) \times (b \mod n)] \mod n = (a \times b) \mod n \]
  - \((-1) \mod n = n - 1\)
    - (Using \(b = q \cdot n + r\), with \(b = -1\), \(q = -1\) and \(r = n-1\))
Brief History

- All encryption algorithms from BC till 1976 were secret key algorithms
  - Also called private key algorithms or symmetric key algorithms
  - Julius Caesar used a substitution cipher
  - Widespread use in World War II (enigma)

- Public key algorithms were introduced in 1976 by Whitfield Diffie and Martin Hellman
Cryptosystem

\((E, D, M, K, C)\)
- \(E\) set of encryption functions
  \(e: M \times K \rightarrow C\)
- \(D\) set of decryption functions
  \(d: C \times K \rightarrow M\)
- \(M\) set of plaintexts
- \(K\) set of keys
- \(C\) set of ciphertexts
Example

- Cæsar cipher
  - $\mathcal{M} = \{ \text{sequences of letters} \}$
  - $\mathcal{K} = \{ i \mid i \text{ is an integer and } 0 \leq i \leq 25 \}$
  - $\mathcal{E} = \{ E_k \mid k \in \mathcal{K} \text{ and for all letters } m, E_k(m) = (m + k) \mod 26 \}$
  - $\mathcal{D} = \{ D_k \mid k \in \mathcal{K} \text{ and for all letters } c, D_k(c) = (26 + c - k) \mod 26 \}$
  - $\mathcal{C} = \mathcal{M}$
**Cæsar cipher**

- Let \( k = 9 \), \( m = \text{“VELVET” (21 4 11 21 4 19)} \)
  - \( E_k(m) = (30 13 20 30 13 28) \mod 26 \)
    - \( = \text{“4 13 20 4 13 2” = “ENUENC”} \)
  - \( D_k(m) = (26 + c - k) \mod 26 \)
    - \( = (21 30 37 21 30 19) \mod 26 \)
    - \( = \text{“21 4 11 21 4 19” = “VELVET”} \)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>N</td>
<td>O</td>
<td>P</td>
<td>Q</td>
<td>R</td>
<td>S</td>
<td>T</td>
<td>U</td>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
</tr>
</tbody>
</table>
Attacks

- **Ciphertext only:**
  - adversary has only $Y$;
  - goal?

- **Known plaintext:**
  - adversary has $X$, $Y$;
  - goal?

- **Chosen plaintext:**
  - adversary gets a specific plaintext enciphered;
  - goal?
Classical Cryptography

Key Source

Encrypt (algorithm)

Alice

Plaintext X

Secure Channel

Secret key K

Decrypt (algorithm)

Ciphertext Y

Bob

Plaintext X

Oscar

X', K'

Ed (Cryptoanalyst)
Classical Cryptography

- Sender, receiver share common key
  - Keys may be the same, or trivial to derive from one another
  - Sometimes called *symmetric cryptography*

- Two basic types
  - Transposition ciphers
  - Substitution ciphers

- Product ciphers
  - Combinations of the two basic types
Classical Cryptography

- \( y = E_k(x) \) : Ciphertext \( \rightarrow \) Encryption
- \( x = D_k(y) \) : Plaintext \( \rightarrow \) Decryption
- \( k \) = encryption and decryption key
- The functions \( E_k() \) and \( D_k() \) must be inverses of one another
  - \( E_k(D_k(y)) = ? \)
  - \( D_k(E_k(x)) = ? \)
  - \( E_k(D_k(x)) = ? \)
Transposition Cipher

- Rearrange letters in plaintext to produce ciphertext
- Example (Rail-Fence Cipher)
  - Plaintext is “HELLO WORLD”
  - Rearrange as
    - HLOOL
    - ELWRD
  - Ciphertext is HLOOL ELWRD
Attacking the Cipher

- Anagramming
  - If 1-gram frequencies match English frequencies, but other $n$-gram frequencies do not, probably transposition
  - Rearrange letters to form $n$-grams with highest frequencies
Example

- **Ciphertext:** HLOOLELWRD
- Frequencies of 2-grams beginning with H
  - HE 0.0305
  - HO 0.0043
  - HL, HW, HR, HD < 0.0010
- Frequencies of 2-grams ending in H
  - WH 0.0026
  - EH, LH, OH, RH, DH ≤ 0.0002
- Implies E follows H
Example

- Arrange so that H and E are adjacent

  HE
  LL
  OW
  OR
  LD

- Read off across, then down, to get original plaintext
Substitution Ciphers

- Change characters in plaintext to produce ciphertext
- Example (Caesar cipher)
  - Plaintext is HELLO WORLD;
  - Key is 3, usually written as letter ‘D’
  - Ciphertext is KHOOR ZRUOG
Attacking the Cipher

- **Brute Force: Exhaustive search**
  - If the key space is small enough, try all possible keys until you find the right one
  - Cæsar cipher has 26 possible keys

- **Statistical analysis**
  - Compare to 1-gram model of English
Statistical Attack

- Ciphertext is KHOOR ZRUOG
- Compute frequency of each letter in ciphertext:
  
<table>
<thead>
<tr>
<th>Letter</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>0.1</td>
</tr>
<tr>
<td>H</td>
<td>0.1</td>
</tr>
<tr>
<td>K</td>
<td>0.1</td>
</tr>
<tr>
<td>O</td>
<td>0.3</td>
</tr>
<tr>
<td>R</td>
<td>0.2</td>
</tr>
<tr>
<td>U</td>
<td>0.1</td>
</tr>
<tr>
<td>Z</td>
<td>0.1</td>
</tr>
</tbody>
</table>

- Apply 1-gram model of English
  - Frequency of characters (1-grams) in English is on next slide
# Character Frequencies (Denning)

<table>
<thead>
<tr>
<th>a</th>
<th>0.080</th>
<th>h</th>
<th>0.060</th>
<th>n</th>
<th>0.070</th>
<th>t</th>
<th>0.090</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>0.015</td>
<td>i</td>
<td>0.065</td>
<td>o</td>
<td>0.080</td>
<td>u</td>
<td>0.030</td>
</tr>
<tr>
<td>c</td>
<td>0.030</td>
<td>j</td>
<td>0.005</td>
<td>p</td>
<td>0.020</td>
<td>v</td>
<td>0.010</td>
</tr>
<tr>
<td>d</td>
<td>0.040</td>
<td>k</td>
<td>0.005</td>
<td>q</td>
<td>0.002</td>
<td>w</td>
<td>0.015</td>
</tr>
<tr>
<td>e</td>
<td>0.130</td>
<td>l</td>
<td>0.035</td>
<td>r</td>
<td>0.065</td>
<td>x</td>
<td>0.005</td>
</tr>
<tr>
<td>f</td>
<td>0.020</td>
<td>m</td>
<td>0.030</td>
<td>s</td>
<td>0.060</td>
<td>y</td>
<td>0.020</td>
</tr>
<tr>
<td>g</td>
<td>0.015</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>z</td>
<td>0.002</td>
</tr>
</tbody>
</table>
Statistical Analysis

- $f(c)$: frequency of character $c$ in ciphertext
- $\varphi(i)$:
  - correlation of frequency of letters in ciphertext with corresponding letters in English, assuming key is $i$
  - $\varphi(i) = \sum_{0 \leq c \leq 25} f(c)p(c - i)$
  - so here,
    - $\varphi(i) = 0.1p(6 - i) + 0.1p(7 - i) + 0.1p(10 - i) + 0.3p(14 - i) + 0.2p(17 - i) + 0.1p(20 - i) + 0.1p(25 - i)$
      - $p(x)$ is frequency of character $x$ in English
  - Look for maximum correlation!
Correlation: $\varphi(i)$ for $0 \leq i \leq 25$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$\varphi(i)$</th>
<th>$i$</th>
<th>$\varphi(i)$</th>
<th>$i$</th>
<th>$\varphi(i)$</th>
<th>$i$</th>
<th>$\varphi(i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0482</td>
<td>7</td>
<td>0.0442</td>
<td>13</td>
<td>0.0520</td>
<td>19</td>
<td>0.0315</td>
</tr>
<tr>
<td>1</td>
<td>0.0364</td>
<td>8</td>
<td>0.0202</td>
<td>14</td>
<td>0.0535</td>
<td>20</td>
<td>0.0302</td>
</tr>
<tr>
<td>2</td>
<td>0.0410</td>
<td>9</td>
<td>0.0267</td>
<td>15</td>
<td>0.0226</td>
<td>21</td>
<td>0.0517</td>
</tr>
<tr>
<td>3</td>
<td>0.0575</td>
<td>10</td>
<td>0.0635</td>
<td>16</td>
<td>0.0322</td>
<td>22</td>
<td>0.0380</td>
</tr>
<tr>
<td>4</td>
<td>0.0252</td>
<td>11</td>
<td>0.0262</td>
<td>17</td>
<td>0.0392</td>
<td>23</td>
<td>0.0370</td>
</tr>
<tr>
<td>5</td>
<td>0.0190</td>
<td>12</td>
<td>0.0325</td>
<td>18</td>
<td>0.0299</td>
<td>24</td>
<td>0.0316</td>
</tr>
<tr>
<td>6</td>
<td>0.0660</td>
<td></td>
<td></td>
<td>19</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


The Result

- **Ciphertext** is **KHOOR ZRUOG**
- **Most probable keys**, based on $\varphi$:
  - $i = 6$, $\varphi(i) = 0.0660$
    - Plaintext: **EBIIL TLOLA** (How?)
  - $i = 10$, $\varphi(i) = 0.0635$
    - Plaintext: **AXEEH PHKEW** (How?)
  - $i = 3$, $\varphi(i) = 0.0575$
    - Plaintext: **HELLO WORLD** (How?)
  - $i = 14$, $\varphi(i) = 0.0535$
    - Plaintext: **WTAAD LDGAS**
Cæsar’s Problem

- Key is too short
  - Can be found by exhaustive search
  - Statistical frequencies not concealed well
    - They look too much like regular English letters

- So make it longer
  - Multiple letters in key
  - Idea is to smooth the statistical frequencies to make cryptanalysis harder
Vigenère Cipher

- Like Cæsar cipher, but use a phrase
- Example
  - Message THE BOY HAS THE BALL
  - Key VIG
  - Encipher using Cæsar cipher for each letter:
    - key VIGVIGVIGVIGVIGVIGVIG
    - plain THEBOYHASTHEBALL
    - cipher OPKWWECIYOPKWIRG
### Relevant Parts of Tableau

<table>
<thead>
<tr>
<th></th>
<th>G</th>
<th>I</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>G</td>
<td>I</td>
<td>V</td>
</tr>
<tr>
<td>B</td>
<td>H</td>
<td>J</td>
<td>W</td>
</tr>
<tr>
<td>E</td>
<td>K</td>
<td>M</td>
<td>Z</td>
</tr>
<tr>
<td>H</td>
<td>N</td>
<td>P</td>
<td>C</td>
</tr>
<tr>
<td>L</td>
<td>R</td>
<td>T</td>
<td>G</td>
</tr>
<tr>
<td>O</td>
<td>U</td>
<td>W</td>
<td>J</td>
</tr>
<tr>
<td>S</td>
<td>Y</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>T</td>
<td>Z</td>
<td>B</td>
<td>O</td>
</tr>
<tr>
<td>Y</td>
<td>E</td>
<td>H</td>
<td>T</td>
</tr>
</tbody>
</table>

- Tableau with relevant rows, columns only
- **Example encipherments:**
  - key V, letter T: follow V column down to T row (giving “O”)
  - Key I, letter H: follow I column down to H row (giving “P”)

Useful Terms

- **period**: length of key
  - In earlier example, period is 3
- **tableau**: table used to encipher and decipher
  - Vigènere cipher has key letters on top, plaintext letters on the left
- **polyalphabetic**: the key has several different letters
  - Cæsar cipher is monoalphabetic
Attacking the Cipher

- Key to attacking vigenère cipher
  - determine the key length
  - If the keyword is n, then the cipher consists of n monoalphabetic substitution ciphers

```
key   VIGVIGVIGVIGVIGV
plain THEBOYHASTHEBALL
cipher OPKWWEICYOPKWIRG
```

```
key   DECEPTIVEDECEPTIVEDECEPTIVE
plain WEAREDISCOVEREDSAVEYOURSELF
cipher ZICVTVQNGRZGVTVWAVZHCQYGLMGJ
```
One-Time Pad

- A Vigenère cipher with a random key at least as long as the message
  - Provably unbreakable; Why?
  - Consider ciphertext DXQR. Equally likely to correspond to
    - plaintext DOIT (key AJIY) and
    - plaintext DONT (key AJDY) and any other 4 letters
  - Warning: keys *must* be random, or you can attack the cipher by trying to regenerate the key
    - Approximations, such as using pseudorandom number generators to generate keys, are *not* random
Overview of the DES

- A block cipher:
  - encrypts blocks of 64 bits using a 64 bit key
  - outputs 64 bits of ciphertext
  - A product cipher
    - performs both substitution and transposition (permutation) on the bits
    - basic unit is the bit
- Cipher consists of 16 rounds (iterations) each with a round key generated from the user-supplied key
Round keys are 48 bits each
- Extracted from 64 bits
- Permutation applied

Deciphering involves using round keys in reverse
Encipherment

\[ \begin{align*}
L_0 &\oplus f(K_1) = R_0 \\
R_1 &= L_0 \oplus f(R_0, K_1) \\
L_{16} &= R_{15} \\
R_{16} &= L_{15} \oplus f(R_{15}, K_{16})
\end{align*} \]
The \( f \) Function

\[ \text{Expansion} \]

\[ R_{f^{-1}} \text{ (32 bits)} \]

\[ \oplus \]

\[ \text{S-box} \]

\[ \text{S1} \quad \text{S2} \quad \text{S3} \quad \text{S4} \quad \text{S5} \quad \text{S6} \quad \text{S7} \quad \text{S8} \]

\[ K_i \text{ (48 bits)} \]

6 bits into each

4 bits out of each

32 bits
Controversy

- Considered too weak
  - Design to break it using 1999 technology published
  - Design decisions not public
    - S-boxes may have backdoors
- Several other weaknesses found
  - Mainly related to keys
DES Modes

- Electronic Code Book Mode (ECB):
  - Encipher each block independently

- Cipher Block Chaining Mode (CBC)
  - XOR each block with previous ciphertext block
  - Uses an initialization vector for the first one

\[
\begin{align*}
\text{init. vector} & \quad m_1 \quad \oplus \quad m_2 \quad \oplus \quad \ldots \\
\text{DES} & \quad c_1 \quad \text{sent} \\
\text{DES} & \quad c_2 \quad \text{sent} \\
\end{align*}
\]
CBC Mode Decryption

CBC has self healing property
- If one block of ciphertext is altered, the error propagates for at most two blocks
Self-Healing Property

- Initial message
  - 3231343336353837 3231343336353837
  - 3231343336353837 3231343336353837

- Received as (underlined 4c should be 4b)
  - ef7c4cb2b4ce6f3b f6266e3a97af0e2c
  - 746ab9a6308f4256 33e60b451b09603d

- Which decrypts to
  - efca61e19f4836f1 3231333336353837
  - 3231343336353837 3231343336353837

- Incorrect bytes underlined; plaintext “heals” after 2 blocks
Public Key Cryptography

- Two keys
  - *Private key* known only to individual
  - *Public key* available to anyone

- Idea
  - Confidentiality:
    - encipher using public key,
    - decipher using private key
  - Integrity/authentication:
    - encipher using private key,
    - decipher using public one
Requirements

1. Given the appropriate key, it must be computationally easy to encipher or decipher a message

2. It must be computationally infeasible to derive the private key from the public key

3. It must be computationally infeasible to determine the private key from a chosen plaintext attack
Diffie-Hellman

- Compute a common, shared key
  - Called a *symmetric key exchange protocol*
- Based on discrete logarithm problem
  - Given integers \( n \) and \( g \) and prime number \( p \), compute \( k \) such that \( n = g^k \mod p \)
  - Solutions known for small \( p \)
  - Solutions computationally infeasible as \( p \) grows large – hence, choose large \( p \)
Algorithm

- Constants known to participants
  - prime $p$; integer $g$ other than 0, 1 or $p$–1
- Alice: (private = $k_A$, public = $K_A$)
- Bob: (private = $k_B$, public = $K_B$)
  - $K_A = g^{k_A} \mod p$
  - $K_B = g^{k_B} \mod p$
- To communicate with Bob,
  - Alice computes $S_{A,B} = K_B^{k_A} \mod p$
- To communicate with Alice,
  - Bob computes $S_{B,A} = K_A^{k_B} \mod p$

$S_{A,B} = S_{B,A} ?$
Example

- Assume $p = 53$ and $g = 17$
- Alice chooses $k_A = 5$
  - Then $K_A = 17^5 \mod 53 = 40$
- Bob chooses $k_B = 7$
  - Then $K_B = 17^7 \mod 53 = 6$
- Shared key:
  - $K_B^{k_A} \mod p = 6^5 \mod 53 = 38$
  - $K_A^{k_B} \mod p = 40^7 \mod 53 = 38$

Exercise:
Let $p = 5$, $g = 3$
$k_A = 4, k_B = 3$

$K_A = ?, K_B = ?, S = ?$,
RSA

- Relies on the difficulty of determining the number of numbers relatively prime to a large integer $n$
- **Totient** function $\phi(n)$
  - Number of $+\,$ integers less than $n$ and relatively prime to $n$
- Example: $\phi(10) = 4$
  - What are the numbers relatively prime to 10?
  - $\phi(77)\,$?
  - $\phi(p)\,$? When $p$ is a prime number
  - $\phi(pq)\,$? When $p$ and $q$ are prime numbers
Algorithm

- Choose two large prime numbers $p, q$
  - Let $n = pq$; then $\phi(n) = (p-1)(q-1)$
  - Choose $e < n$ relatively prime to $\phi(n)$.
  - Compute $d$ such that $ed \mod \phi(n) = 1$
    - Public key: $(e, n)$;
    - private key: $(d, n)$
- Encipher: $c = m^e \mod n$
- Decipher: $m = c^d \mod n$
Confidentiality using RSA
Authentication using RSA

Message Source → Encryption → Decryption → Message Source

Alice → Y → Bob

Key Source
Confidentiality + Authentication

Message Source → Encryption → Encryption → Decryption → Decryption → Message Source


Key Source
Warnings

- Encipher message in blocks considerably larger than the examples here
  - If 1 character per block, RSA can be broken using statistical attacks (just like classical cryptosystems)
  - Attacker cannot alter letters, but can rearrange them and alter message meaning
    - Example: reverse enciphered message: ON to get NO
Cryptographic Checksums

- Mathematical function to generate a set of $k$ bits from a set of $n$ bits (where $k \leq n$).
  - $k$ is smaller than $n$ except in unusual circumstances
  - Keyed CC: requires a cryptographic key
    \[ h = C_{Key}(M) \]
  - Keyless CC: requires no cryptographic key
    - Message Digest or One-way Hash Functions
      \[ h = H(M) \]
- Can be used for message authentication
  - Hence, also called Message Authentication Code (MAC)
Mathematical characteristics

- Every bit of the message digest function potentially influenced by every bit of the function’s input
- If any given bit of the function’s input is changed, every output bit has a 50 percent chance of changing
- Given an input file and its corresponding message digest, it should be computationally infeasible to find another file with the same message digest value
Definition

- Cryptographic checksum function $h: A \to B$:
  1. For any $x \in A$, $h(x)$ is easy to compute
     - Makes hardware/software implementation easy
  2. For any $y \in B$, it is computationally infeasible to find $x \in A$ such that $h(x) = y$
     - One-way property
  3. It is computationally infeasible to find $x, x' \in A$ such that $x \neq x'$ and $h(x) = h(x')$
  4. Alternate form: Given any $x \in A$, it is computationally infeasible to find a different $x' \in A$ such that $h(x) = h(x')$. 
Collisions

- If \( x \neq x' \) and \( h(x) = h(x') \), \( x \) and \( x' \) are a collision
  - Pigeonhole principle: if there are \( n \) containers for \( n+1 \) objects, then at least one container will have 2 objects in it.
  - Application: suppose \( n = 5 \) and \( k = 3 \). Then there are 32 elements of A and 8 elements of B, so
    - each element of B has at least 4 corresponding elements of A
Keys

- Keyed cryptographic checksum: requires cryptographic key
  - DES in chaining mode: encipher message, use last $n$ bits. Requires a key to encipher, so it is a keyed cryptographic checksum.

- Keyless cryptographic checksum: requires no cryptographic key
  - MD5 and SHA-1 are best known; others include MD4, HAVAL, and Snefru
Message Digest

- MD2, MD4, MD5 (Ronald Rivest)
  - Produces 128-bit digest;
  - MD2 is probably the most secure, longest to compute (hence rarely used)
  - MD4 is a fast alternative; MD5 is modification of MD4

- SHA, SHA-1 (Secure Hash Algorithm)
  - Related to MD4; used by NIST’s Digital Signature
  - Produces 160-bit digest
  - SHA-1 may be better

- SHA-256, SHA-384, SHA-512
  - 256-, 384-, 512 hash functions designed to be use with the Advanced Encryption Standards (AES)

- Example:
  - MD5(There is $1500 in the blue box) = f80b3fde8ecbac1b515960b9058de7a1
  - MD5(There is $1500 in the blue box) = a4a5471a0e019a4a502134d38fb64729
Hash Message Authentication Code (HMAC)

- Make keyed cryptographic checksums from keyless cryptographic checksums
- \( h \) be keyless cryptographic checksum function that takes data in blocks of \( b \) bytes and outputs blocks of \( l \) bytes. \( k' \) is cryptographic key of length \( b \) bytes (from \( k \))
  - If short, pad with 0s’ to make \( b \) bytes; if long, hash to length \( b \)
- \( ipad \) is 00110110 repeated \( b \) times
- \( opad \) is 01011100 repeated \( b \) times
- HMAC-\( h(k, m) = h(k' \oplus opad || h(k' \oplus ipad || m)) \)
  - \( \oplus \) exclusive or, \( || \) concatenation
Protection Strength

- Unconditionally Secure
  - Unlimited resources + unlimited time
  - Still the plaintext CANNOT be recovered from the ciphertext

- Computationally Secure
  - Cost of breaking a ciphertext exceeds the value of the hidden information
  - The time taken to break the ciphertext exceeds the useful lifetime of the information
## Average time required for exhaustive key search

<table>
<thead>
<tr>
<th>Key Size (bits)</th>
<th>Number of Alternative Keys</th>
<th>Time required at $10^6$ Decryption/µs</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>$2^{32} = 4.3 \times 10^9$</td>
<td>2.15 milliseconds</td>
</tr>
<tr>
<td>56</td>
<td>$2^{56} = 7.2 \times 10^{16}$</td>
<td>10 hours</td>
</tr>
<tr>
<td>128</td>
<td>$2^{128} = 3.4 \times 10^{38}$</td>
<td>$5.4 \times 10^{18}$ years</td>
</tr>
<tr>
<td>168</td>
<td>$2^{168} = 3.7 \times 10^{50}$</td>
<td>$5.9 \times 10^{30}$ years</td>
</tr>
</tbody>
</table>
Key Points

- Two main types of cryptosystems: classical and public key
- Classical cryptosystems encipher and decipher using the same key
  - Or one key is easily derived from the other
- Public key cryptosystems encipher and decipher using different keys
  - Computationally infeasible to derive one from the other