Encryption

A Brief Overview

• Encryption:
  – Definition: mechanisms to disguise the message so that if the information is intercepted/diverted, the content of the message will not be understood.
  – Impact: foundational building block to security-based computing

Terminology

• Scenario:
  – $S$ wants to send the message $T$ to $R$, where an outsider, $O$, wants the message and tries to access it.
  – $S$: Sender
  – $R$: Receiver
  – $T$: Transmission Medium
  – $O$: Interceptor or Intruder.
• 4 ways $O$ might try to access message.
  – Block it: prevent $T$ from reaching $R$ (availability)
  – Intercept it: read or listen to message (secrecy)
  – Modify it: obtaining message and changing it
  – Fabricate: generate an authentic-looking message to be delivered to $R$ appearing to come from $S$
**Terminology**

- **Encryption**: process of encoding a message so that its meaning is not obvious
- **Decryption**: transforming encrypted message back to its normal form
- **Encode/decode**: translating phrases to other words or phrases
- **Encipher/decipher**: translating letters or symbols individually.
- **Plaintext**: original form of message: \( P = (p_1, p_2, \ldots, p_n) \)
- **Ciphertext**: encrypted form of message: \( C = (c_1, c_2, \ldots, c_n) \)
- **Encryption/decryption relationships**:
  - \( C = E(P); P = D(C); P = D(E(P)) \)

**Encryption Algorithms**

- Some encryption algs use a key \( K \)
  - \( C = E(K, P) \)
  - \( E \) is a SET of encryption algs
  - Key \( K \) selects specific one
- **Symmetric Encryption**: \( P = D(K, E(K, P)) \)
  - encryption/decryption keys are the same
- **Asymmetric Encryption**: \( P = D(K_D, E(K_E, P)) \)

**Pictorial Representation**

**Symmetric Encryption:**

- Key
- Plaintext
- Encryption
- Ciphertext
- Decryption
- Original Plaintext

**Asymmetric Encryption:**

- Encryption Key \( K_E \)
- Decryption Key \( K_D \)
- Plaintext
- Encryption
- Ciphertext
- Decryption
- Original Plaintext
More Terms

- Cryptography: (hidden writing)
  - Practice of using encryption to conceal text
- Cryptanalyst:
  - Person who studies encryption and encrypted messages
  - Intent: find hidden meaning
- Cryptographer and Cryptanalyst:
  - Both attempt to translate coded material to original form
  - Cryptographer: works on behalf of legitimate sender or receiver.
  - Cryptanalyst: Works on behalf of unauthorized interceptor
- Cryptology: research/study into encryption/decryption
  - Includes cryptography and cryptanalysis.

Cryptanalysis

- Objective: Break an encryption
  - Deduce the meaning of a ciphertext msg
  - Determine decrypting algorithm that matches an encrypting algorithm
- Possible techniques:
  - break single message
  - Recognize patterns in encrypted msgs
  - break subsequent msgs with straightforward decryption alg
  - Find general weaknesses in encryption alg
  - Without necessarily intercepting any msgs
- Tools:
  - Encrypted msgs, known encryption algs, intercepted plaintext, data elements known/suspected of being in ciphertext, mathematical/statistical techniques, pros of languages, computers, and luck

Breakable Encryption

- Encryption algorithm is BREAKABLE:
  - Given enough time/data, an analyst could determine alg.
  - Practicality is issue
  - For given cipher scheme, may have $10^{30}$ possible decipherments
  - Select one from $10^{30}$
  - Current technology: perform $10^{10}$ ops/sec
    - Require $10^{20}$ secs == $10^{12}$ years
- Reality Check:
  - Cryptanalyst won’t just try the “hard” ways
    - Ex: more clever approach, might only take $10^{15}$ ops
    - $10^{10}$/ops/sec, $10^{15}$ ops will take about one day
  - Breakability estimates are based on CURRENT technology
Character Representations

- Study ways to encrypt any computer material:
  - ASCII/EBCDIC chars
  - Binary data or Object code
  - Control stream

<table>
<thead>
<tr>
<th>ABCD</th>
<th>EFHIJKL</th>
<th>MNOPQRST</th>
<th>UVWXYZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>0123</td>
<td>456789</td>
<td>101112</td>
<td>131415</td>
</tr>
</tbody>
</table>

Substitution-based Encryption

- Monoalphabetic Ciphers
  - Caesar Cipher: $c_i = E(p_i) = p_i + 3$
  - wkhphvdjh lv qrw wrr kdug wr euhdn
  - Easy to perform in field (no written instructions)
  - Permutation: reordering of the elements
    - $c_i = a_{\pi}(p_i)$; $\pi(\lambda) = 25 - \lambda$
    - Use a key:

<table>
<thead>
<tr>
<th>ABCDEF</th>
<th>GHIJKL</th>
<th>MNOPQRST</th>
<th>UVWXYZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEY</td>
<td>ABCDEF</td>
<td>GHIJKL</td>
<td>MNOPQRST</td>
</tr>
<tr>
<td>SPEACT</td>
<td>LHRDF</td>
<td>GHIEJK</td>
<td>NQOVW</td>
</tr>
</tbody>
</table>

  - Weakness: study frequency distribution

Polyalphabetic Substitution Ciphers

- Desire flat distribution
- Combine distributions that are high with low ones
  - Encipher $T$ as $a$ and sometimes as $b$
  - Also encipher $X$ as $a$ and sometimes as $b$
- Use two separate encryption alphabets
  - Tables for odd and even positions
    - $a_1(\lambda) = (3 \times \lambda) \mod 26$
    - $a_2(\lambda) = (5 \times \lambda + 13) \mod 26$
    - TREAT YIMPO SSIBL E
    - Fumir dyft czysz h
Substitution Discussion

- Major weakness:
  - Frequency distribution
    - (Index of coincidence: measure of variation between frequencies in a distribution)
  - Some letters are used more frequently than others
  - Numerous enciphering techniques still make it difficult to hide these patterns
- Kasiski Method: find number of alphabets used
  - Identify repeated patterns of 3 or more characters
  - For each pattern, write down position at which each instance of pattern begins
  - Compute difference between start points of success instances
  - Determine all factors of each difference
  - If polyalphabetic substitution used, key length will be one of the factors that appears often in previous steps.

Transpositions (Permutations)

- Definition: encryption where letters are rearranged.
- Goal: diffusion, spread info from message or key out widely across the ciphertext.
- Try to break established patterns.

Transposition Techniques

- Columnar Transpositions:
  - Rearrangement of chars of plaintext into cols

C1  C2  C3  C4  C5
C6  C7  C8  C9  C10
C11 C12 Etc.

THIS IS
SAME S
SAGE T
SHOW O
HAW C
OLUMN
ART RA
POSITION
ITION
WORKS

tsosu oaniw haaso lristo inghw utpir seeoa nrook istwc nasns
Transpositions

- **Digram**: patterns of adjacent letters.
  - Study 2 and 3 letter combinations of adj letters
- **Double Transposition Alg**:
  - Involves 2 columnar transpositions
  - With different number of columns, applied sequentially
- **Fractionated Morse**:
  - Keyed monoalphabetic cipher
  - Result is subsequently blocked (clustered)
  - Morse code is used as its basis

Secure Encryption Systems

- Previous algs could be completed manually, although tedious
  - Decryption could also be done manually
- New technology requires more “hard” encryption algs to hinder cryptanalysts
- Review 3 key, important encryption algs
- Look at recent developments

Important Encryption Algs

- **Merkle-Hellman knapsack**:
  - Alg based on “hard” problems (NP-complete)
- **Rivest-Shamir-Adelman (RSA)**:
  - More resilient to attacks than Merkle alg
- **Data Encryption Standard (DES)**:
  - Developed with support from NIST
  - Provide secure encryption for commercial applications
- **Clipper program**:
  - Skipjack: cryptographic alg – maintain secrecy
Some “Hard” theories

- **NP-complete:**
  - Encryption algos that would require NP-complete alg to decrypt
- **Number theory:**
  - Inverses
  - Primes
  - Modular Arithmetic
  - Euclidean alg: procedure for computing gcd of 2 numbers.

Public Key Encryption

- **Traditional key system:**
  - Need a key for every pair of users
  - \( N(N-1)/2 \) keys, grows exponentially with users
  - Each user has to keep track of many keys
- **Public key (asymmetric encryption system):**
  - Each user has 2 keys: public and private key
  - May publish the public key freely, inverses
  - \( P=D(k_{PRIV}, E(k_{PUB}, P)) \)
  - Only 2 keys are needed per user
  - B, C, and D can ally encrypt mesgs for A with A’s public key

Merkle-Hellman Knapsacks

- **Knapsack problem:**
  - Set of positive integers
  - Target sum
  - Find subset of integers that equal the target
  - NP-complete alg.
- **Encode binary mesg as soln to knapsack problem**
  - Reduce ciphertext to target sum
  - By adding terms corresponding to 1s in plaintext
  - Convert blocks of plaintext to knapsack sum by adding into sum the terms that match with 1 bits in plaintext.
Superincreasing Knapsack

- Superincreasing sequence:
  - Each integer is greater than sum of all preceding integers
  - \( s_k > \sum_{j=1}^{k-1} s_j \)
  - Solution of superincreasing knapsack (e.g., simple knapsack) is easy to find
- Convert simple knapsack into Hard knapsack
  - Pick superincreasing sequence \( S \) of \( m \) integers
    - \( S = [s_1, s_2, ..., s_m] \)
  - Choose multiplier \( w \) and modulus \( n \), \( n > \sum_{j=1}^{m-1} s_j \)
  - Choose \( n \) to be prime
  - Replace every \( s_i \) in simple knapsack with term:
    \[ h_i = w * s_i \mod n \]
  - Hard knapsack: \( H = [h_1, h_2, ..., h_m] \)

Merkle-Hellman (cont’d)

- Merkle-Hellman is Public key cryptosystem
  - Each user has public key:
    - Set of integers of a (simple) knapsack problem
  - Each user has private key
    - Set of integers for corresponding superincreasing knapsack
- Contribution: design of technique to convert superincreasing knapsack into a regular one.
  - Change numbers in nonobvious, reversible way.

Merkle-Hellman (cont’d)

- Encryption alg starts with binary message
  - \( P = [p_1, p_2, ..., p_k] \)
  - Divide message into blocks of \( m \) bits, \( P_0 = [p_1, p_2, ..., p_m], P_1 = [p_1, p_2, ..., P_{2m}] \).
    - Value of \( m \) is number of terms in simple or hard knapsack
  - Encipherment of message \( P \) is sequence of targets
    - Each target is sum of some of the terms of the hard knapsack \( H \)
    - Terms selected correspond to 1 bits in \( P_i \)
    - \( P_i \) serves as selection vector for elts of \( H \)
    - Each term of ciphertext is \( P_i \cdot H \)
Merkle-Hellman (cont’d)

- Decryption:
  - Legitimate recipient knows simple knapsack and values of \( w \) and \( n \)
  \[
  H = w \cdot S \mod n
  \]
  \[
  C = H \cdot P = w \cdot S \cdot P \mod n
  \]
- To decipher, multiply \( C \) by \( w^{-1} \)
  \[
  w^{-1} \cdot C = w^{-1} \cdot H \cdot P = w^{-1} \cdot w \cdot S \cdot P = S \cdot P \mod n
  \]

- Weaknesses:
  - How easy is it to determine \( w \) or \( n \) from \( H \)?

Example

- \( S = \{1,2,4,9\}; H = \{15,13,9,16\} \)
- \( w = 15, n = 17, m = 4; h = w \cdot s \mod n \)
- \( P = 0100101110100101 \)
- Encode with \( H \) as follows:
  - \( [0,1,0,0] \cdot [15,13,9,16] = 13 \)
  - \( [1,0,1,1] \cdot [15,13,9,16] = 40 \)
  - \( [1,0,1,0] \cdot [15,13,9,16] = 24 \)
  - \( [0,1,0,1] \cdot [15,13,9,16] = 29 \)
- Encrypted message as integers: 13,40,24,29,
  - Public knapsack \( H = \{15,13,9,16\} \)

RSA: Rivest-Shamir-Adelman

- Superficially looks similar to Merkle-Hellman:
- Exploits number theory and finding prime factors of a target:
  \[
  C = P^e \mod n; P = C^d \mod n
  \]
- Symmetry in modular arithmetic
  - encryption/decryption are mutual inverses and commutative.
  \[
  P = C^d \mod n = (P^e)^d \mod n = (P^d)^e \mod n
  \]
- Choosing keys: \( (e, n) \) and \( (d,n) \)
  - Select value for \( n \)
    - should be quite large: a product of two large primes \( p \) and \( q \) (100 digits ea)
  - Select value for \( e \): relatively prime to \( (p-1) \cdot (q-1) \)
    - \( E \) has no common factors with above product.
    - Choose \( e \) as prime larger than both \( (p-1) \) and \( (q-1) \)
  - Select value for \( d \): \( e \cdot d \equiv 1 \mod (p-1) \cdot (q-1) \)
- How to use: user distributes \( e \) and \( n \), keeps \( d \) secret
- To encrypt, need to find large prime numbers
DES: Data Encryption Standard

- Developed for US govt for general public use.
- Repeats 16 cycles of substitution and transposition
  - Shannon’s theory of information secrecy
    - Confusion: info is changed so that output bits have no obvious relation to input bits
    - Diffusion: spread the effect of one plaintext bits to other ciphertext bits.
  - Splits data block into 2 pieces:
    - Scrambles each half independently
    - Combines key with one half
      - (key is transformed during each cycle)
    - Swap 2 halves
    - Repeat 16 times.

One Cycle in DES

[Pfleeger97]