CSE331: Introduction to Networks and Security

Lecture 18
Fall 2002
Announcements

• Prof. Zdancewic will be out of town Nov. 5th – 14th
• Lectures will be covered by Prof. Gunter
• Aditya Chadha (course TA) will cover projects
Recap

• Cracking Polyalphabetic Ciphers
  – Kasiski Method of determining key length
  – Index of Coincidence

• Today:
  – Industrial Strength Encryption
  – Shared Key Cryptography
Kinds of Industrial Strength Crypto

- Shared Key Cryptography
- Public Key Cryptography
- Cryptographic Hashes
Shared Key Cryptography

• Sender & receiver use the same key
• Key must remain private
• Also called *symmetric or secret key cryptography*
• Often are *block-ciphers*
  – Process plaintext data in blocks
• Examples: DES, Triple-DES, Blowfish, Twofish, Rijndael, …
Shared Key Notation

- Encryption algorithm
  \[ E : \text{key} \times \text{plain} \rightarrow \text{cipher} \]
  Notation: \( K\{\text{msg}\} = E(K, \text{msg}) \)

- Decryption algorithm
  \[ D : \text{key} \times \text{cipher} \rightarrow \text{plain} \]
  D inverts E
  \[ D(K, E(K, \text{msg})) = \text{msg} \]

- Use capital “K” for shared (secret) keys

- Sometimes E is the same algorithm as D
Secure Channel: Shared Keys

Alice

K_{AB}{Hello!} \rightarrow K_{AB}{Hi!}

Bart

K_{AB}
Problems with Shared Key Crypto

- Compromised key means interceptors can decrypt any ciphertext they’ve acquired.
  - Change keys frequently to limit damage
- Distribution of keys is problematic
  - Keys must be transmitted securely
  - Use couriers?
  - Distribute in pieces over separate channels?
- Number of keys is $O(n^2)$ where $n$ is # of participants
- Potentially easier to break?
Data Encryption Standard (DES)

• Adopted as a standard in 1976
• Security analyzed by the National Security Agency (NSA)
• Key length is 56 bits
  – padded to 64 bits by using 8 parity bits
• Uses simple operators on (up to) 64 bit values
  – Simple to implement in software or hardware
• Input is processed in 64 bit blocks
• Based on a series of 16 rounds
  – Each cycle uses permutation & substitution to combine plaintext with the key
DES Encryption

INPUT

INITIAL PERMUTATION

PERMUTED INPUT

L_0

R_0

K_1

L_1=R_0

R_1=L_0+f(R_0, K_1)

K_2

L_2=R_1

R_2=L_1+f(R_1, K_2)

K_n

L_15=R_{14}

R_{15}=L_{14}+f(R_{14}, K_{15})

K_{11}

R_{16}=L_{15}+f(R_{15}, K_{16})

L_{16}=R_{15}

INVERSE INITIAL PERM

OUTPUT
One Round of DES (f of previous slide)

- Expansion
- Permutation choice of key
- S-box
- Permutation
Types of Permutations in DES

- Permutation
- Permuted Choice
- Expansion Permutation
DES S-Boxes

- Substitution table
- 6 bits of input replaced by 4 bits of output
- Which substitution is applied depends on the input bits

- Implemented as a lookup table
  - 8 S-Boxes
  - Each S-Box has a table of 64 entries
  - Each entry specifies a 4-bit output
DES Decryption

• Use the same algorithm as encryption, but use $k_{16} \ldots k_1$ instead of $k_1 \ldots k_{16}$

• Proof that this works:
  – To obtain round $j$ from $j-1$:
    1. $L_j = R_{j-1}$
    2. $R_j = L_{j-1} \oplus f(R_{j-1}, k_j)$

  – Rewrite in terms of round $j-1$:
    1. $R_{j-1} = L_j$
    2. $L_{j-1} \oplus f(R_{j-1}, k_j) = R_j$
    $L_{j-1} \oplus f(R_{j-1}, k_j) \oplus f(R_{j-1}, k_j) = R_j \oplus f(R_{j-1}, k_j)$
    $L_{j-1} = R_j \oplus f(R_{j-1}, k_j)$
    $L_{j-1} = R_j \oplus f(L_j, k_j)$
Problems with DES

- Key length too short: 56 bits
  - [www.distributed.net](http://www.distributed.net) broke a DES challenge in 1999 in under 24 hours (parallel attack)

- Other problems
  - Bit-wise complementation of key produces bit-wise complemented ciphertext
  - Not all keys are good (half 0’s half 1’s)
  - Differential cryptanalysis: Carefully choose pairs of plaintext that differ in particular known ways (I.e. they are
## Block Cipher Performance

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Key Length</th>
<th>Block Size</th>
<th>Rounds</th>
<th>Clks/Byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>Twofish</td>
<td>variable</td>
<td>128</td>
<td>16</td>
<td>18.1</td>
</tr>
<tr>
<td>Blowfish</td>
<td>variable</td>
<td>64</td>
<td>16</td>
<td>19.8</td>
</tr>
<tr>
<td>Square</td>
<td>128</td>
<td>128</td>
<td>8</td>
<td>20.3</td>
</tr>
<tr>
<td>RC5-32/16</td>
<td>variable</td>
<td>64</td>
<td>32</td>
<td>24.8</td>
</tr>
<tr>
<td>CAST-128</td>
<td>128</td>
<td>64</td>
<td>16</td>
<td>29.5</td>
</tr>
<tr>
<td>DES</td>
<td>56</td>
<td>64</td>
<td>16</td>
<td>43</td>
</tr>
<tr>
<td>Serpent</td>
<td>128, 192, 256</td>
<td>128</td>
<td>32</td>
<td>45</td>
</tr>
<tr>
<td>SAFER (S)K-128</td>
<td>128</td>
<td>64</td>
<td>8</td>
<td>52</td>
</tr>
<tr>
<td>FEAL-32</td>
<td>64, 128</td>
<td>64</td>
<td>32</td>
<td>65</td>
</tr>
<tr>
<td>IDEA</td>
<td>128</td>
<td>64</td>
<td>8</td>
<td>74</td>
</tr>
<tr>
<td>Triple-DES</td>
<td>112</td>
<td>64</td>
<td>48</td>
<td>116</td>
</tr>
</tbody>
</table>
Advanced Encryption Standard (AES)

- National Institute of Standards & Technology (NIST)
  - Computer Security Research Center (CSRC)

- Uses the Rijndael algorithm
  - Invented by Belgium researchers
    Dr. Joan Daemen & Dr. Vincent Rijmen
  - Adopted May 26, 2002
  - Key length: 128, 192, or 256 bits
  - Block size: 128, 192, or 256 bits
Public Key Cryptography

- Sender encrypts using a *private key*
- Receiver decrypts using a *public key*
- Only the private key must be kept secret
  - Public key can be distributed at will
- Also called *asymmetric* cryptography
- Can be used for digital signatures
- Examples: RSA, El Gamal, DSA
- Usually more expensive than secret key cryptography
Public Key Notation

- Encryption algorithm
  \( E : \text{keyPub} \times \text{plain} \rightarrow \text{cipher} \)
  Notation: \( K\{msg\} = E(K, \text{msg}) \)

- Decryption algorithm
  \( D : \text{keyPriv} \times \text{cipher} \rightarrow \text{plain} \)
  Notation: \( k\{msg\} = D(k,\text{msg}) \)

- \( D \) inverts \( E \)
  \( D(k, E(K, \text{msg})) = \text{msg} \)

- Use capital “\( K \)” for public keys
- Use lower case “\( k \)” for private keys

- Occasionally \( E \) is the same algorithm as \( D \)
Secure Channel: Private Key

Alice

K_A, K_B
k_A

K_B{Hello!}

K_A{Hi!}

Bart

K_A, K_B
k_B
Cryptographic Hashes

- Creates a hard-to-invert summary of input data
- Useful for integrity properties
  - Sender computes the hash of the data, transmits data and hash
  - Receiver uses the same hash algorithm, checks the result
- Like a check-sum or error detection code
  - Uses a cryptographic algorithm internally
  - More expensive to compute
- Sometimes called a Message Digest
- Examples:
  - Secure Hash Algorithm (SHA)
  - Message Digest (MD4, MD5)