Lecture 2: Cryptography I

CS 336/536: Computer Network Security
Fall 2015

Nitesh Saxena

Course Administration

• Everyone receiving my emails?
• Lecture slides worked okay?
  – Both ppt and pdf versions
• Everyone knows how to access the course web page?
• I am posting the lectures in advance (the morning before the lecture)
  – But, this should not affect the attendance 😊
Course Admin

• No labs this week (labs not active yet)

• We will do a 10-min break around 6:15pm
  – Please remind/shout in case I forget

Outline of today’s lecture

• Cryptography Overview
• Private Key Cryptography: Encryption
• Classical Ciphers
• Block Cipher -- DES
Cryptography

- Etymology: Secret (Crypt) Writing (Graphy)
- Study of mathematical techniques to achieve various goals in information security, such as confidentiality, authentication, integrity, non-repudiation, etc.
- Not the only means of providing network security, rather a subset of techniques.
- Quite an old field!

Cryptography: Cast of Characters

- Alice (A) and Bob (B): communicating parties
- Eve (E): Eavesdropping (or passive) adversary
- Mallory (M): Man-in-the-Middle (or active adversary)
- Trent (T): a trusted third party (TTP)
Today’s Focus

• How to achieve confidentiality by means of cryptography?

Private Key/Public Key Cryptography

• **Private Key**: Sender and receiver share a common (private) key
  – Encryption and Decryption is done using the private key
  – Also called conventional/shared-key/single-key/symmetric-key cryptography

• **Public Key**: Every user has a private key and a public key
  – Encryption is done using the public key and Decryption using private key
  – Also called two-key/asymmetric-key cryptography
Common Terminologies

- Plaintext
- Key
- Encrypt (encipher)
- Ciphertext
- Decrypt (decipher)
- Cipher
- Cryptosystem
- Cryptanalysis (codebreaking)
- Cryptology: Cryptography + Cryptanalysis

Private key model

[Diagram showing the private key model with plaintext input and ciphertext output]

9/2/2015 Lecture 2 - Cryptography - 1

9/2/2015 Lecture 2 - Cryptography - 1
Open vs Closed Design

- Closed Design (as was followed in military communication during the World Wars)
  - Keep the cipher secret
  - Also sometimes referred to as the “proprietary design”
  - Bad practice! (why?)

- Open Design (*Kerckhoffs’ principle*)
  - Keep everything public, except the key
  - Good practice – this is what we focus upon!

Private Key Encryption: main functions

1. KeyGen: $K = \text{KeyGen}(l)$ ($l$ is a security parameter)

2. Enc: $C = \text{Enc}(K,M)$

3. Dec: $M = \text{Dec}(K,C)$
Goals of the Attacker

- Learn the plaintext corresponding to a given ciphertext — **One-Way Security**
- Extract the key — **Key Recovery Security**
- Learn some information about the plaintext corresponding to a given ciphertext — **Semantic Security**
- *Key recovery security and one-way security are a must for an encryption scheme. Semantic Security is ideal.*

Capabilities of the Attacker

1. **No Information** (besides the algorithm)
2. **Ciphertext only**
   - Adversary knows only the ciphertext(s)
3. **Known plaintext**
   - Adversary knows a set of plaintext-ciphertext pairs
4. **Chosen (and adaptively chosen) plaintext** (CPA attack)
   - Adversary chooses a number of plaintexts and obtains the corresponding ciphertexts
5. **Chosen (and adaptively chosen) ciphertext attack** (CCA attack)
   - Adversary chooses a number of ciphertexts and obtains the corresponding plaintexts
Security Model

least attacker capability ........................................ most attacker capability

1<2<3<4<5

weakest cryptosystem ........................................ strongest cryptosystem

• 1 is the hardest and 5 is the easiest attack to perform
• A cryptosystem secure against 5 is the strongest, and secure against 1 is the weakest
• A cryptosystem secure against 5 is automatically secure against 4, 3, 2 and 1

Brute Force Attacks: Key Recovery

• Since the key space is finite, given a pair (or more) of plaintext and ciphertext, a cryptanalyst can try and check all possible keys.
• For above to be not feasible, key space should be large!!
  – How large?
  – Large enough to make it impractical for an adversary. But what is impractical today, may not be so tomorrow. At least $2^{80}$ – see this paper on “selecting cryptographic key sizes”
Ciphers We Will Study

• Classical ones
  – Substitution Ciphers
    • Caesar’s Cipher
    • Monoalphabetic
    • Polyalphabetic
  – Transposition Ciphers

• Modern ones
  – DES/AES
  – Others...

Caesar Cipher (or Shift Cipher)

• Substitution cipher
• Let messages be all lower case from a through z (no spaces or punctuation).
• Represent letters by numbers from 0 to 25.
• Encryption function
  \[ C_i = E(P_i) = P_i + K \pmod{26} \]
  where \( K \) is secret key
• Decryption is
  \[ P_i = D(C_i) = C_i - K \pmod{26} \]
Security of Caesar Cipher

- Easy to brute force: size of key-space is 26
  - Not secure against even ciphertext-only attack
    (the one where adversary had the least capability)

Monoalphabetic Substitution

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

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

- Key space is large $26! = 4 \times 10^{26}$
  - Quite large, however,
  - Can be broken (not secure against ciphertext-only) using language characteristics!

Polyalphabetic Substitution – Vigenere Cipher

- Use $K$ mono-alphabetic ciphers – $E_1, E_2, \ldots, E_K$.
- In position $i$, of plaintext, use cipher $E_i$.
- Example using Caesar ciphers ...

Plaintext: helloi loveyouwontyou tellmyouname
Key: polytechnicpolytechnicpolytechnicpoly
Ciphertext: wswijhmnv………………………………

- A little harder to break but frequency analysis is possible
- Some well known techniques for determining key length – we will not cover (see text for Kasiski method)
One time Pad or Vernam Cipher: Best Possible Cipher

- If we use Vigenere with key length as long as plaintext, then cryptanalysis will be difficult!
- If we change key every time we encrypt then cryptanalyst’s job becomes even more difficult. **One-time pad** or **Vernam Cipher**.
- How do we get such long keys?
- Such a cipher is difficult to break but not very practical.

**Binary Vernam**

- plaintext is binary string and key is binary string of equal length, then encryption can be done by a simple XOR operation.

| Plaintext: | 01010000010001010011 |
| Key:       | 110101001001100111 |
| Ciphertext:| 10000101011000110100 |

- **If the key is random and is not re-used**, then such a system offers unconditional security – perfect secrecy!
- Intuitively perfect secrecy can be seen from the fact that given any plaintext and ciphertext, there is a key which maps the selected plaintext to the selected ciphertext. So given a ciphertext, we get no information whatsoever on what key or plaintext could have been used.
- How do we obtain “random” bit-strings for shared secret keys as long as the messages, and never re-use them?
- Again system is **not practical**.
Transposition

- Harder to break than substitution ciphers
- Still susceptible to frequency analysis

Product Ciphers

- Substitution and transposition ciphers are not secure due to language characteristics
- What about using two or more of these ciphers in a serial fashion
  - Two or more substitutions
  - Two or more Transpositions
  - A few substitutions and a few transposition
  - Transition from classical to modern ciphers
Some Questions

- Enigma is an example of how design?
- Encryption can provide confidentiality, but not integrity: true or false?
- World’s best cipher is what?
- I give you a ciphertext, and ask you to give me the corresponding plaintext – what attack is this? How does it compare to the known plaintext attack?
- All classical ciphers are based on either or why are they all broken?
- What’s the problem in choosing a long long key? It should give you a lot of security, no?

Some Questions

- An encryption scheme is said to be deterministic if encrypting the same plaintext twice yields the same ciphertext. (otherwise it is said to be randomized).
  – Is a deterministic scheme a good scheme in terms of security?
Further Reading

- Stallings (edition 5) – Chapter 2.1 to 2.3
- HAC – Chapter 1 and 7

Today’s fun/informative bit – The Smudge Attack

- See: http://www.usenix.org/event/woot10.tech/full_papers/Aviv.pdf
Block Ciphers and Stream Ciphers

- Block ciphers partition plaintext into blocks and encrypt each block independently (with the same key) to produce ciphertext blocks.
- A stream cipher generates a *keystream* and encrypts by combining the keystream with the plaintext, usually with the bitwise XOR operation.
- We will focus mostly on Block Ciphers

DES – Data Encryption Standard

- Encrypts by series of substitution and transpositions.
- Based on *Feistel Structure*
- Worldwide standard for more than 20 years.
- Designed by IBM (Lucifer) with later help from NSA.
- No longer considered secure for highly sensitive applications.
- Replacement standard AES (advanced encryption standard) recently completed.
DES – Overview (Block Operation)

DES – Each Round
DES – Function F

DES – Key Schedule (KS)
## Operation Tables of DES: Key Schedule, PC-1, PC-2

<table>
<thead>
<tr>
<th>Location (n)</th>
<th>No. of shifts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
</tr>
</tbody>
</table>

## Key permutation PC-1

<table>
<thead>
<tr>
<th>Key</th>
<th>17</th>
<th>11</th>
<th>23</th>
<th>22</th>
<th>17</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>56</td>
<td>30</td>
<td>42</td>
<td>34</td>
<td>26</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>59</td>
<td>24</td>
<td>41</td>
<td>43</td>
<td>27</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>60</td>
<td>54</td>
<td>36</td>
<td>52</td>
<td>20</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>62</td>
<td>46</td>
<td>58</td>
<td>24</td>
<td>30</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>63</td>
<td>58</td>
<td>30</td>
<td>52</td>
<td>14</td>
<td>31</td>
</tr>
<tr>
<td>6</td>
<td>64</td>
<td>52</td>
<td>36</td>
<td>24</td>
<td>30</td>
<td>14</td>
</tr>
</tbody>
</table>

## Key permutation PC-2

<table>
<thead>
<tr>
<th>Key</th>
<th>17</th>
<th>11</th>
<th>23</th>
<th>22</th>
<th>17</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>59</td>
<td>24</td>
<td>41</td>
<td>43</td>
<td>27</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>56</td>
<td>30</td>
<td>42</td>
<td>34</td>
<td>26</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>60</td>
<td>54</td>
<td>36</td>
<td>52</td>
<td>20</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>62</td>
<td>46</td>
<td>58</td>
<td>24</td>
<td>30</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>63</td>
<td>58</td>
<td>30</td>
<td>52</td>
<td>14</td>
<td>31</td>
</tr>
<tr>
<td>6</td>
<td>64</td>
<td>52</td>
<td>36</td>
<td>24</td>
<td>30</td>
<td>14</td>
</tr>
</tbody>
</table>

## Operation Tables (IP, IP⁻¹, E and P)

### Initial Permutation (IP)

| 38 | 55 | 42 | 31 | 48 | 57 | 2 |
| 46 | 52 | 34 | 32 | 18 | 16 | 28 |
| 47 | 51 | 37 | 33 | 17 | 27 | 29 |
| 34 | 50 | 43 | 30 | 49 | 56 | 19 |
| 40 | 59 | 41 | 39 | 45 | 54 | 21 |
| 58 | 66 | 55 | 44 | 33 | 22 | 10 |

### Bit-Selection Table E

| 37 | 1 | 5 | 2 | 3 | 4 | 3 |
| 4 | 3 | 6 | 7 | 8 | 9 |
| 8 | 9 | 10 | 11 | 12 | 13 |
| 12 | 13 | 14 | 15 | 16 | 17 |
| 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | 32 | 33 | 34 | 35 |

### Inverse Initial Permutation (IP⁻¹)

| 38 | 1 | 16 | 6 | 26 | 18 | 57 | 42 |
| 46 | 17 | 11 | 14 | 27 | 30 | 28 | 55 |
| 47 | 15 | 10 | 13 | 29 | 25 | 31 | 32 |
| 34 | 50 | 43 | 30 | 49 | 56 | 19 |
| 40 | 59 | 41 | 39 | 45 | 54 | 21 |
| 58 | 66 | 55 | 44 | 33 | 22 | 10 |

### Permutation P

| 9 | 7 | 30 | 21 |
| 19 | 12 | 8 | 17 |
| 15 | 33 | 24 |
| 5 | 18 | 31 | 10 |
| 2 | 9 | 14 | 18 |
| 17 | 37 | 7 | 5 |
| 19 | 13 | 30 | 9 |
| 55 | 58 | 8 | 54 |
S-boxes: S1 (as an example)

<table>
<thead>
<tr>
<th></th>
<th>0000</th>
<th>0001</th>
<th>0010</th>
<th>0011</th>
<th>0100</th>
<th>0101</th>
<th>0110</th>
<th>0111</th>
<th>1000</th>
<th>1001</th>
<th>1010</th>
<th>1011</th>
<th>1100</th>
<th>1101</th>
<th>1110</th>
<th>1111</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>14</td>
<td>4</td>
<td>13</td>
<td>1</td>
<td>2</td>
<td>15</td>
<td>11</td>
<td>8</td>
<td>3</td>
<td>10</td>
<td>6</td>
<td>12</td>
<td>5</td>
<td>9</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>15</td>
<td>7</td>
<td>4</td>
<td>14</td>
<td>2</td>
<td>13</td>
<td>1</td>
<td>10</td>
<td>6</td>
<td>12</td>
<td>11</td>
<td>9</td>
<td>5</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>1</td>
<td>14</td>
<td>8</td>
<td>13</td>
<td>6</td>
<td>2</td>
<td>11</td>
<td>15</td>
<td>12</td>
<td>9</td>
<td>7</td>
<td>3</td>
<td>10</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>15</td>
<td>12</td>
<td>8</td>
<td>2</td>
<td>4</td>
<td>9</td>
<td>1</td>
<td>7</td>
<td>5</td>
<td>11</td>
<td>3</td>
<td>14</td>
<td>10</td>
<td>0</td>
<td>6</td>
<td>13</td>
</tr>
</tbody>
</table>

Is the table entry from

$$S(b_1b_2b_3b_4b_5b_6) \quad \text{row: } b_1b_2 \quad \text{column: } b_3b_4b_5b_6$$

$$S(011001) = 6_d = 0110$$

---

DES Decryption

- Same as the encryption algorithm with the “reversed” key schedule – NEXT!
Plain text

Initial permutation (IP)

Round-1 (key $K_1$)

Rounds 2-15

Round-16 (key $K_{16}$)

swap

IP inverse

Cipher text

Since $b \oplus b = 0$

$\begin{align*}
R_{15} & \quad L_{15} \\
L_{15} \oplus F(R_{15}, K_{16}) & \quad R_{15} \\
& \quad L_{15} \oplus F(R_{15}, K_{16}) \oplus F(R_{15}, K_{16}) \\
& \quad R_{15} \\
& \quad L_{15} \\
& \quad R_{15}
\end{align*}$
DES Example

We choose a random plaintext block and a random key, and determine what the ciphertext block would be (all in hexadecimal):

Plaintext: 123456ABCD132536
CipherText: C0B7A8D05F3A829C
Key: AABB09182736CCDD

<table>
<thead>
<tr>
<th>Round</th>
<th>Left</th>
<th>Right</th>
<th>Round Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Round 1</td>
<td>18CA18AD</td>
<td>5A78E394</td>
<td>194CD072DE8C</td>
</tr>
<tr>
<td>Round 2</td>
<td>5A78E394</td>
<td>4A1210F6</td>
<td>4568581ABCCD</td>
</tr>
<tr>
<td>Round 3</td>
<td>4A1210F6</td>
<td>B8089591</td>
<td>06ED4A5CF5B5</td>
</tr>
<tr>
<td>Round 4</td>
<td>B8089591</td>
<td>236779C2</td>
<td>DA2D032BEE3</td>
</tr>
</tbody>
</table>

Example (contd) -- encryption

| Round 5 | 236779C2 | A15A4B87 | 59A629FEC913 |
| Round 6 | A15A4B87 | 2E8F9C65 | C194B87475E |
| Round 7 | 2E8F9C65 | A9FC20A3 | 708AD2DDB3C0 |
| Round 8 | A9FC20A3 | 308BEE97 | 34F822F0C66D |
| Round 9 | 308BEE97 | 10AF9D37 | 34B4473DCCC |
| Round 10 | 10AF9D37 | 6C6A5CB0 | 32765708B5BF |
| Round 11 | 6C6A5CB0 | FF3C405F | 5D5560AF7CA5 |
| Round 12 | FF3C405F | 22A5963B | C2C1E36A4BF3 |
| Round 13 | 22A5963B | 387CCDA4 | 9C31397C91F |
| Round 14 | 387CCDA4 | BD2D02AB | 251B58BC717D0 |
| Round 15 | BD2D02AB | CF26B472 | 3330C59A36D |
| Round 16 | 19BA9212 | CF26B472 | 181C5D75C66D |

After combination: 19BA9212CF26B472
CipherText: C0B7A8D05F3A829C
(after final permutation)
Example (contd) -- decryption

Let us see how Bob, at the destination, can decipher the ciphertext received from Alice using the same key. Table 6.16 shows some interesting points.

<table>
<thead>
<tr>
<th>Ciphertext: C0B7A8D05F3A829C</th>
</tr>
</thead>
<tbody>
<tr>
<td>After initial permutation: 19BA9212CF26B472</td>
</tr>
<tr>
<td>After splitting: L0=19BA9212  R0=CF26B472</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Round</th>
<th>Left</th>
<th>Right</th>
<th>Round Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Round 1</td>
<td>CF26B472</td>
<td>BD2DD2AB</td>
<td>181C5D75C66D</td>
</tr>
<tr>
<td>Round 2</td>
<td>BD2DD2AB</td>
<td>387CCDAA</td>
<td>3330C5D9A36D</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Round 15</td>
<td>5A78E394</td>
<td>18CA18AD</td>
<td>4568581ABCCE</td>
</tr>
<tr>
<td>Round 16</td>
<td>14A7D678</td>
<td>18CA18AD</td>
<td>194CD072DE8C</td>
</tr>
</tbody>
</table>

After combination: 14A7D67818CA18AD
Plaintext: 123456ABCD132536 (after final permutation)

DES Security: Avalanche Effect

Plaintext: 0000000000000000
Ciphertext: 4789FD476E82A5F1
Key: 22234512987ABB23

Plaintext: 0000000000000001
Ciphertext: 0A4ED5C15A63FEA3
Key: 22234512987ABB23

Although the two plaintext blocks differ only in the rightmost bit, the ciphertext blocks differ in 29 bits. This means that changing approximately 1.5 percent of the plaintext creates a change of approximately 45 percent in the ciphertext.
Further Reading

- Chapter 7.4 of HAC
- Chapter 3 of Stallings

DES Security

- S-Box design not well understood
- Has survived some recent sophisticated attacks (differential cryptanalysis)
- Key is too short. Hence is vulnerable to brute force attack.
- 1998 distributed attack took 3 months.
- $1,000,000 machine will crack DES in 35 minutes – 1997 estimate. $10,000 – 2.5 days.
Super-encryption.

- If key length is a concern, then instead of encrypting once, encrypt twice!!
  
  \[ C = E_{K2}(E_{K1}(P)) \]
  
  \[ P = D_{K1}(D_{K2}(C)) \]

- Does this result in a larger key space?
- Encrypting with multiple keys is known as super-encryption.
- May not always be a good idea
Double DES

- Double DES is almost as easy to break as single DES (Needs more memory though)!

\[
\begin{align*}
P & \xrightarrow{K_1} E \xrightarrow{X} E \xrightarrow{K_2} C \\
C & \xrightarrow{K_2} D \xrightarrow{X} D \xrightarrow{K_1} P
\end{align*}
\]

Double DES – Meet-in-the-middle Attack (due to Diffie-Hellman)

- Based on the observation that, if
  \[ C = E_{K_2}(E_{K_1}(P)) \]
  Then
  \[ X = E_{K_1}(P) = D_{K_2}(C). \]
- Given a known (P, C) pair, encrypt P with all possible values of K and store result in table T.
- Next, decrypt C with all possible keys K and check result. If match occurs then check key pair with new known (P, C) pair. If match occurs, you have found the keys. Else continue as before.
- Process will terminate successfully.
Meet-in-the-middle Explanation

- The first match does not say anything as we have $2^{64}$ ciphertexts and $2^{112}$ keys.
- On the average $2^{112} / 2^{64} = 2^{48}$ keys will produce same ciphertext.
- So there could be $2^{48}$ possible candidates.
- We can use a second pair $(P', C')$.
- So, probability that false alarm will survive two known $(P, C)$ pairs is $2^{48} / 2^{64} = 2^{-16}$.
- One can always check a third pair to further reduce the chance of a false alarm.

Triple DES

- Triple DES (2 keys) requires $2^{112}$ search. Is reasonably secure.
- Triple DES (3 keys) requires $2^{112}$ as well.
- Which one is better?
Block Cipher Encryption modes

- Electronic Code Book (ECB)
- Cipher Block Chain (CBC)
  - Most popular one
- Others (we will not cover)
  - Cipher Feed Back (CFB)
  - Output Feed Back (OFB)

Analysis

We will analyze each of these modes in terms of:
- Security
- Computational Efficiency (parallelizing encryption/decryption)
- Transmission Errors
Electronic Code Book (ECB) Mode

- Although DES encrypts 64 bits (a block) at a time, it can encrypt a long message (file) in Electronic Code Book (ECB) mode.

- Deterministic -- If same key is used then identical plaintext blocks map to identical ciphertext

Example – why ECB is bad?

Tux

Tux encrypted with AES in ECB mode
Cipher Block Chain (CBC) Mode

CBC Traits

- Randomized encryption
- IV – Initialization vector serves as the randomness for first block computation; the ciphertext of the previous block serves as the randomness for the current block computation
- IV is a random value
- IV is **no secret**; it is sent along with the ciphertext blocks (it is part of the ciphertext)
Example – why CBC is good?

CBC – More Properties

• What happens if $k$-th cipher block $C_k$ gets corrupted in transmission.
  – With ECB – Only decrypted $P_k$ is affected.
  – With CBC?
    • Only blocks $P_k$ and $P_{k+1}$ are affected!!
• What if one plaintext block $P_k$ is changed?
  – With ECB only $C_k$ affected.
  – With CBC all subsequent ciphertext blocks will be affected.
    • “Avalanche effect”
  – This leads to an effective integrity protection mechanism (or message authentication code (MAC))
Some Questions

• Double encryption in DES increases the key space size from $2^{56}$ to $2^{112}$ – true or false?
• Is known-plaintext an active or a passive attack?
• Is chosen-ciphertext attack an active or a passive attack?
• Reverse Engineering is applied to what design of systems – open or closed?

Some Questions

• $C=\text{DES}(K,P)$; where $(P, C$ are 64-bit long blocks). What would be $\text{DES}(K,"PPPP")$ in ECB mode? What it would be in CBC mode?
• ECB is secure for sending just one block of data: true or false?
• Is it okay to re-use IV in CBC? Why/why not?
• Alice needs to send a *long* top-secret message to Bob. Which of the ciphers that we studied today can she use?