All the afternoon Mungo had been working on Stern's code, principally with the aid of the latest messages which he had copied down at the Nevin Square drop. Stern was very confident. He must be well aware London Central knew about that drop. It was obvious that they didn't care how often Mungo read their messages, so confident were they in the impenetrability of the code.

—Talking to Strange Men, Ruth Rendell

Outline

• will consider:
  – Block vs stream ciphers
  – Feistel cipher design & structure
  – DES
    • details
    • strength
  – Differential & linear cryptanalysis
  – Block cipher design principles

Modern Block Ciphers

• now look at modern block ciphers
• one of the most widely used types of cryptographic algorithms
• provide secrecy/authentication services
• Initially look at Data Encryption Standard (DES) to illustrate block cipher design issues
Block vs Stream Ciphers

- Block ciphers process messages in blocks, each of which is then en/decrypted
- Like a substitution on very big characters
  - 64-bits or more (these days, 128 or more)
- Stream ciphers process messages a bit or byte at a time when en/de-crypting
- Many current ciphers are block ciphers
  - Better analysed
  - Broader range of applications

Block Cipher Principles

- Many symmetric block ciphers are based on a **Feistel Cipher Structure** (more below)
- Feistel makes it possible to decrypt ciphertext efficiently to recover messages
- Block ciphers look like an extremely large substitution
- For a 64-bit block would need table of $2^{64}$ entries
- This $2^{64}$ entry table would be the **key**
- Instead create from smaller building blocks, using idea of a product cipher, ...
  - And a much smaller key
Claude Shannon and Substitution-Permutation Ciphers

- Claude Shannon introduced idea of substitution-permutation (S-P) networks in 1949 paper
- form basis of modern block ciphers
- S-P nets are based on the two primitive cryptographic operations seen before:
  - substitution (S-box)
  - permutation (P-box)
- provide confusion & diffusion (respectively) of message & key

Confusion and Diffusion

- cipher needs to completely obscure statistical properties of original message
- a one-time pad does this
- more practically Shannon suggested combining S and P elements to obtain:
  - confusion (S) – makes relationship between ciphertext and key as complex as possible
  - diffusion (P) – dissipates statistical structure of plaintext over bulk of ciphertext

Feistel Cipher Structure

- Horst Feistel devised the **Feistel Cipher**
  - based on concept of invertible product cipher
- partitions input block into two halves
  - process through multiple rounds which
  - perform a substitution on left data half
  - based on round function of right half & subkey
  - then have permutation swapping halves
- implements Shannon’s S-P net concept
Feistel Cipher Design Elements

- block size
- key size
- number of rounds
- subkey generation algorithm
- round function
- fast software en/decryption
- ease/difficulty of analysis

Data Encryption Standard (DES)

- (used to be the) most widely used block cipher in world
- adopted in 1977 by NBS (now NIST)
  - as FIPS PUB 46
- encrypts 64-bit data using 56-bit key
- has widespread use
- has been considerable controversy over its security (esp. size of key)

DES History

- IBM developed Lucifer cipher
  - by team led by Feistel in late 60’s
  - used 64-bit data blocks with 128-bit key
- then redeveloped as a commercial cipher with input from NSA and others
- in 1973 NBS issued request for proposals for a national cipher standard
- IBM submitted their revised Lucifer which was eventually accepted as the DES
DES Design Controversy

- although DES standard is public
- was considerable controversy over design
  - in choice of 56-bit key (vs. Lucifer’s 128-bit)
  - and because design criteria were classified
- subsequent events and public analysis show in fact design was appropriate
- use of DES flourished (from 80s to 00s)
  - especially in financial applications
  - still standardised for legacy application use

DES Encryption Overview

- Initial Permutation IP
  - first step of the data computation
  - IP reorders the input data bits
  - even bits to LH half, odd bits to RH half
  - quite regular in structure (easy in h/w)
  - example:
    \[
    \text{IP}(675a6967 \ 5e5a6b5a) = (ffb2194d \ 004df6fb)
    \]
- DES Round Structure
  - uses two 32-bit L & R halves
  - as for any Feistel cipher can describe it as:
    \[
    L_i = R_{i-1} \quad R_i = L_{i-1} \oplus F(R_{i-1}, K_i)
    \]
  - \(F\) takes 32-bit R half and 48-bit subkey:
    - expands R to 48-bits using perm E
    - adds to subkey using XOR
    - passes through 8 S-boxes to get 32-bit result
    - finally permutes using 32-bit perm P
Substitution Boxes S

- have eight S-boxes which map 6 to 4 bits
- each S-box is actually 4 little 4 bit boxes
  - outer bits 1 & 6 (row bits) select one row of 4
  - inner bits 2-5 (col bits) are substituted
- result is 8 lots of 4 bits, or 32 bits
- row selection depends on both data & key
  - feature known as autoclaving (autokeying)
- example:
  - $S(18 \ 09 \ 12 \ 3d \ 11 \ 17 \ 38 \ 39) = 5fd25e03$
DES Key Schedule

- forms subkeys used in each round
  - initial permutation of the key (PC1) which selects 56-bits in two 28-bit halves
  - 16 stages consisting of:
    - rotating each half separately either 1 or 2 places depending on the key rotation schedule K
    - selecting 24-bits from each half & permuting them by PC2 for use in round function F
  - note practical use issues in h/w vs. s/w

DES Decryption

- decrypt must unwind steps of data computation
- with Feistel design, do encryption steps again using subkeys in reverse order (SK16 ... SK1)
  - IP undoes final FP step of encryption
  - 1st round with SK16 undoes 16th encrypt round
  - ...
  - 16th round with SK1 undoes 1st encrypt round
  - then final FP undoes initial encryption IP
  - thus recovering original data value
Avalanche Effect

- key desirable property of any encryption algorithm
- want a change of one input or key bit to result in changing approx half output bits
- making attempts to “home-in” by guessing keys impossible
- DES exhibits strong avalanche

<table>
<thead>
<tr>
<th>Round</th>
<th>$K_1$</th>
<th>$L_1$</th>
<th>$K_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>5a05ba0</td>
<td>3cf0ac0f</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1e83f0538bd2930</td>
<td>3cf03cf0f</td>
<td>bad2845</td>
</tr>
<tr>
<td>2</td>
<td>0a123432232131</td>
<td>bad2245</td>
<td>98678923</td>
</tr>
<tr>
<td>3</td>
<td>23456789989098</td>
<td>98908721</td>
<td>bade3309</td>
</tr>
<tr>
<td>4</td>
<td>0234567823421620</td>
<td>00203920e</td>
<td>82419469</td>
</tr>
<tr>
<td>5</td>
<td>3233433438662c05</td>
<td>42418469</td>
<td>1b9f5a41</td>
</tr>
<tr>
<td>6</td>
<td>1234604964242ae1</td>
<td>1b9f5a41</td>
<td>98678923</td>
</tr>
<tr>
<td>7</td>
<td>0213210c130611</td>
<td>98678923</td>
<td>8717c7f2</td>
</tr>
<tr>
<td>8</td>
<td>1c13072a2832020b</td>
<td>67117c2f</td>
<td>c1bfc09</td>
</tr>
<tr>
<td>9</td>
<td>04292a380634103e</td>
<td>c1bfc09</td>
<td>878f0edc</td>
</tr>
<tr>
<td>10</td>
<td>2739312678050403</td>
<td>887f0edc</td>
<td>10076eb</td>
</tr>
<tr>
<td>11</td>
<td>2826305021261941</td>
<td>60df7e0b</td>
<td>1096540e</td>
</tr>
<tr>
<td>12</td>
<td>120742110400080</td>
<td>5f85b50e</td>
<td>7385385b</td>
</tr>
<tr>
<td>13</td>
<td>30943539301d0d0e</td>
<td>7385385b</td>
<td>6a6e240e</td>
</tr>
<tr>
<td>14</td>
<td>311e09231231182a</td>
<td>c68e20ce</td>
<td>35b0bd75</td>
</tr>
<tr>
<td>15</td>
<td>283e2e0227072520</td>
<td>560bd75</td>
<td>7fe84df</td>
</tr>
<tr>
<td>16</td>
<td>29210013140225</td>
<td>7fe84df</td>
<td>21604490</td>
</tr>
</tbody>
</table>

DES Example

- key desirable property of any encryption algorithm
- want a change of one input or key bit to result in changing approx half output bits
- making attempts to “home-in” by guessing keys impossible
- DES exhibits strong avalanche

Table 3.6 Avalanche Effect in DES: Change in Plaintext

<table>
<thead>
<tr>
<th>Round</th>
<th>$m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0246885e00a86420</td>
<td>1</td>
</tr>
<tr>
<td>1246885e00a86420</td>
<td>1</td>
</tr>
<tr>
<td>3cf30c0fbd32845</td>
<td>1</td>
</tr>
<tr>
<td>3cf30c0fbd32845</td>
<td>1</td>
</tr>
<tr>
<td>bad2284599w9b73c</td>
<td>5</td>
</tr>
<tr>
<td>bad32845939ab7a3</td>
<td>5</td>
</tr>
<tr>
<td>99eb7230b3eb9be</td>
<td>18</td>
</tr>
<tr>
<td>99eb7230b3eb9be</td>
<td>18</td>
</tr>
<tr>
<td>0ba3eb9e2415649</td>
<td>54</td>
</tr>
<tr>
<td>0ba3eb9e2415649</td>
<td>54</td>
</tr>
<tr>
<td>17bdcbb3ccac55a</td>
<td>57</td>
</tr>
<tr>
<td>17bdcbb3ccac55a</td>
<td>57</td>
</tr>
<tr>
<td>42615649180f461</td>
<td>37</td>
</tr>
<tr>
<td>42615649180f461</td>
<td>37</td>
</tr>
<tr>
<td>6b3f9a2b375758ef0f</td>
<td>16</td>
</tr>
<tr>
<td>6b3f9a2b375758ef0f</td>
<td>16</td>
</tr>
<tr>
<td>9616fe23617170c2f</td>
<td>32</td>
</tr>
<tr>
<td>9616fe23617170c2f</td>
<td>32</td>
</tr>
<tr>
<td>cf402c682b200efbc</td>
<td>33</td>
</tr>
<tr>
<td>cf402c682b200efbc</td>
<td>33</td>
</tr>
</tbody>
</table>

Table 3.7 Avalanche Effect in DES: Change in Key

<table>
<thead>
<tr>
<th>Round</th>
<th>$m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0246885e00a86420</td>
<td>0</td>
</tr>
<tr>
<td>0246885e00a86420</td>
<td>0</td>
</tr>
<tr>
<td>3cf30c0fbd32845</td>
<td>3</td>
</tr>
<tr>
<td>3cf30c0fbd32845</td>
<td>3</td>
</tr>
<tr>
<td>bad2284599w9b73c</td>
<td>31</td>
</tr>
<tr>
<td>bad32845939ab7a3</td>
<td>31</td>
</tr>
<tr>
<td>99eb7230b3eb9be</td>
<td>25</td>
</tr>
<tr>
<td>99eb7230b3eb9be</td>
<td>25</td>
</tr>
<tr>
<td>0ba3eb9e2415649</td>
<td>29</td>
</tr>
<tr>
<td>0ba3eb9e2415649</td>
<td>29</td>
</tr>
<tr>
<td>1bb37ac5f340c1a8a9</td>
<td>30</td>
</tr>
<tr>
<td>1bb37ac5f340c1a8a9</td>
<td>30</td>
</tr>
<tr>
<td>6b3f9a2b375758ef0f</td>
<td>31</td>
</tr>
<tr>
<td>6b3f9a2b375758ef0f</td>
<td>31</td>
</tr>
<tr>
<td>9616fe23617170c2f</td>
<td>32</td>
</tr>
<tr>
<td>9616fe23617170c2f</td>
<td>32</td>
</tr>
<tr>
<td>cf402c682b200efbc</td>
<td>33</td>
</tr>
<tr>
<td>cf402c682b200efbc</td>
<td>33</td>
</tr>
</tbody>
</table>

Table 3.8 Avalanche Effect in DES: Change in Initial Vector

<table>
<thead>
<tr>
<th>Round</th>
<th>$m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0246885e00a86420</td>
<td>0</td>
</tr>
<tr>
<td>0246885e00a86420</td>
<td>0</td>
</tr>
<tr>
<td>3cf30c0fbd32845</td>
<td>3</td>
</tr>
<tr>
<td>3cf30c0fbd32845</td>
<td>3</td>
</tr>
<tr>
<td>bad2284599w9b73c</td>
<td>31</td>
</tr>
<tr>
<td>bad32845939ab7a3</td>
<td>31</td>
</tr>
<tr>
<td>99eb7230b3eb9be</td>
<td>25</td>
</tr>
<tr>
<td>99eb7230b3eb9be</td>
<td>25</td>
</tr>
<tr>
<td>0ba3eb9e2415649</td>
<td>29</td>
</tr>
<tr>
<td>0ba3eb9e2415649</td>
<td>29</td>
</tr>
<tr>
<td>17bdcbb3ccac55a</td>
<td>30</td>
</tr>
<tr>
<td>17bdcbb3ccac55a</td>
<td>30</td>
</tr>
<tr>
<td>42615649180f461</td>
<td>31</td>
</tr>
<tr>
<td>42615649180f461</td>
<td>31</td>
</tr>
<tr>
<td>6b3f9a2b375758ef0f</td>
<td>31</td>
</tr>
<tr>
<td>6b3f9a2b375758ef0f</td>
<td>31</td>
</tr>
<tr>
<td>9616fe23617170c2f</td>
<td>32</td>
</tr>
<tr>
<td>9616fe23617170c2f</td>
<td>32</td>
</tr>
<tr>
<td>cf402c682b200efbc</td>
<td>33</td>
</tr>
<tr>
<td>cf402c682b200efbc</td>
<td>33</td>
</tr>
</tbody>
</table>
Strength of DES – Key Size

- 56-bit keys have $2^{56} = 7.2 \times 10^{16}$ values
- brute force search looks hard
- past advances have shown it is possible
  - in 1997 on Internet in a few months
  - in 1998 on dedicated h/w (EFF) in a few days
  - in 1999 above combined in 22hrs!
  - Nowadays ... can do it on a desktop!
- still, it must be possible to recognize plaintext
- forced consideration of alternatives to DES

Attacks on DES – Timing Attacks

- attacks actual implementation of cipher
- use knowledge of consequences of implementation to derive information about some/all subkey bits
- specifically use fact that calculations can take varying times depending on the value of the inputs to it
- particularly problematic on smartcards

Attacks on DES – Analytic Attacks

- now have several analytic attacks on DES
- these utilise some deep structure of the cipher
  - by gathering information about encryptions
  - can eventually recover some/all of the sub-key bits
  - if necessary then exhaustively search for the rest
- generally these are statistical attacks
  - differential cryptanalysis
  - linear cryptanalysis
  - related key attacks

Differential Cryptanalysis

- one of the most significant (public) advances in cryptanalysis
- known by NSA in 70's ... influenced DES
- Murphy, Biham & Shamir published in 90's
- powerful method to analyse block ciphers
- used to analyse most current block ciphers with varying degrees of success
- DES reasonably resistant to it, cf. Lucifer
## Linear Cryptanalysis

- another recent development
- also a statistical method
- must be iterated over rounds, with decreasing probabilities
- developed by Matsui et al in early 90's
- based on finding linear approximations
- can attack DES with $2^{43}$ known plaintexts
- easier but still in practise infeasible

## DES Design Criteria

- as reported by Coppersmith in 1994
- 7 criteria for S-boxes provide for
  - non-linearity
  - resistance to differential cryptanalysis
  - good confusion
- 3 criteria for permutation P provide for
  - increased diffusion

## Block Cipher Design

Many principles from Feistel in 70s still hold
- number of rounds
  - more is better, make exhaustive search the best attack option
- function f:
  - provides “confusion”, is nonlinear, avalanche
  - issues of how S-boxes are selected
- key schedule
  - complex subkey creation, key avalanche