Clark-Wilson Integrity Model

- Integrity defined by a set of constraints
  - Data in a *consistent* or valid state when it satisfies these
- Example: Bank
  - $D$ today’s deposits, $W$ withdrawals, $YB$ yesterday’s balance, $TB$ today’s balance
  - Integrity constraint: $D + YB - W$
- *Well-formed transaction* move system from one consistent state to another
- Issue: who examines, certifies transactions done correctly?
Entities

• CDIs: constrained data items
  – Data subject to integrity controls
• UDIs: unconstrained data items
  – Data not subject to integrity controls
• IVPs: integrity verification procedures
  – Procedures that test the CDIs conform to the integrity constraints
• TPs: transaction procedures
  – Procedures that take the system from one valid state to another
Certification Rules 1 and 2

CR1  When any IVP is run, it must ensure all CDIs are in a valid state

CR2  For some associated set of CDIs, a TP must transform those CDIs in a valid state into a (possibly different) valid state

- Defines relation *certified* that associates a set of CDIs with a particular TP
- Example: TP balance, CDIs accounts, in bank example
Enforcement Rules 1 and 2

ER1 The system must maintain the certified relations and must ensure that only TPs certified to run on a CDI manipulate that CDI.

ER2 The system must associate a user with each TP and set of CDIs. The TP may access those CDIs on behalf of the associated user. The TP cannot access that CDI on behalf of a user not associated with that TP and CDI.

- System must maintain, enforce certified relation
- System must also restrict access based on user ID (*allowed* relation)
Users and Rules

CR3  The allowed relations must meet the requirements imposed by the principle of separation of duty.

ER3  The system must authenticate each user attempting to execute a TP

- Type of authentication undefined, and depends on the instantiation
- Authentication *not* required before use of the system, but *is* required before manipulation of CDIs (requires using TPs)
Logging

CR4 All TPs must append enough information to reconstruct the operation to an append-only CDI.

– This CDI is the log

– Auditor needs to be able to determine what happened during reviews of transactions
Handling Untrusted Input

CR5 Any TP that takes as input a UDI may perform only valid transformations, or no transformations, for all possible values of the UDI. The transformation either rejects the UDI or transforms it into a CDI.

- In bank, numbers entered at keyboard are UDIs, so cannot be input to TPs. TPs must validate numbers (to make them a CDI) before using them; if validation fails, TP rejects UDI
Separation of Duty In Model

ER4 Only the certifier of a TP may change the list of entities associated with that TP. No certifier of a TP, or of an entity associated with that TP, may ever have execute permission with respect to that entity.

- Enforces separation of duty with respect to certified and allowed relations
Comparison With Requirements

1. Users can’t certify TPs, so CR5 and ER4 enforce this
2. Procedural, so model doesn’t directly cover it; but special process corresponds to using TP
   - No technical controls can prevent programmer from developing program on production system; usual control is to delete software tools
3. TP does the installation, trusted personnel do certification
Comparison With Requirements

4. CR4 provides logging; ER3 authenticates trusted personnel doing installation; CR5, ER4 control installation procedure
   • New program UDI before certification, CDI (and TP) after

5. Log is CDI, so appropriate TP can provide managers, auditors access
   • Access to state handled similarly
Comparison to Biba

• Biba
  – No notion of certification rules; trusted subjects ensure actions obey rules
  – Untrusted data examined before being made trusted

• Clark-Wilson
  – Explicit requirements that actions must meet
  – Trusted entity must certify method to upgrade untrusted data (and not certify the data itself)
UNIX Implementation

- Considered “allowed” relation
  \[(user, TP, \{ CDI \text{ set} \})\]
- Each TP is owned by a different user
  - These “users” are actually locked accounts, so no real users can log into them; but this provides each TP a unique UID for controlling access rights
  - TP is setuid to that user
- Each TP’s group contains set of users authorized to execute TP
- Each TP is executable by group, not by world
CDI Arrangement

• CDIs owned by *root* or some other unique user
  – Again, no logins to that user’s account allowed
• CDI’s group contains users of TPs allowed to manipulate CDI
• Now each TP can manipulate CDIs for single user
Examples

• Access to CDI constrained by user
  – In “allowed” triple, \( TP \) can be any TP
  – Put CDIs in a group containing all users authorized to modify CDI

• Access to CDI constrained by TP
  – In “allowed” triple, \( user \) can be any user
  – CDIs allow access to the owner, the user owning the TP
  – Make the TP world executable
Problems

• 2 different users cannot use same copy of TP to access 2 different CDIs
  – Need 2 separate copies of TP (one for each user and CDI set)
• TPs are setuid programs
  – As these change privileges, want to minimize their number
• *root* can assume identity of users owning TPs, and so cannot be separated from certifiers
  – No way to overcome this without changing nature of *root*
Key Points

• Integrity policies deal with trust
  – As trust is hard to quantify, these policies are hard to evaluate completely
  – Look for assumptions and trusted users to find possible weak points in their implementation

• Biba based on multilevel integrity

• Clark-Wilson focuses on separation of duty and transactions
Overview

• Classical Cryptography
  – Cæsar cipher
  – Vigènere cipher
  – DES

• Public Key Cryptography
  – RSA

• Cryptographic Checksums
  – HMAC
Cryptosystem

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

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

- Opponent whose goal is to break cryptosystem is the *adversary*
  - Assume adversary knows algorithm used, but not key
- Three types of attacks:
  - *ciphertext only*: adversary has only ciphertext; goal is to find plaintext, possibly key
  - *known plaintext*: adversary has ciphertext, corresponding plaintext; goal is to find key
  - *chosen plaintext*: adversary may supply plaintexts and obtain corresponding ciphertext; goal is to find key
Basis for Attacks

• Mathematical attacks
  – Based on analysis of underlying mathematics

• Statistical attacks
  – Make assumptions about the distribution of letters, pairs of letters (digrams), triplets of letters (trigrams), etc.
    • Called models of the language
  – Examine ciphertext, correlate properties with the assumptions.
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
  - Combinations are called *product ciphers*
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 the 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
  – Change each letter to the third letter following it (X goes to A, Y to B, Z to C)
    • Key is 3, usually written as letter ‘D’
  – Ciphertext is KHOOR ZRUOG
Attacking the Cipher

• 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

- Compute frequency of each letter in ciphertext:
  
  \[
  \begin{align*}
  &G \ 0.1 \quad H \ 0.1 \quad K \ 0.1 \quad O \ 0.3 \\
  &R \ 0.2 \quad U \ 0.1 \quad Z \ 0.1 
  \end{align*}
  \]

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

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Character</strong></td>
<td><strong>Frequency</strong></td>
</tr>
<tr>
<td>a</td>
<td>0.080</td>
</tr>
<tr>
<td>b</td>
<td>0.015</td>
</tr>
<tr>
<td>c</td>
<td>0.030</td>
</tr>
<tr>
<td>d</td>
<td>0.040</td>
</tr>
<tr>
<td>e</td>
<td>0.130</td>
</tr>
<tr>
<td>f</td>
<td>0.020</td>
</tr>
<tr>
<td>g</td>
<td>0.015</td>
</tr>
<tr>
<td>h</td>
<td>0.060</td>
</tr>
<tr>
<td>i</td>
<td>0.065</td>
</tr>
<tr>
<td>j</td>
<td>0.005</td>
</tr>
<tr>
<td>k</td>
<td>0.005</td>
</tr>
<tr>
<td>l</td>
<td>0.035</td>
</tr>
<tr>
<td>m</td>
<td>0.030</td>
</tr>
<tr>
<td>n</td>
<td>0.070</td>
</tr>
<tr>
<td>o</td>
<td>0.080</td>
</tr>
<tr>
<td>p</td>
<td>0.020</td>
</tr>
<tr>
<td>q</td>
<td>0.002</td>
</tr>
<tr>
<td>r</td>
<td>0.065</td>
</tr>
<tr>
<td>s</td>
<td>0.060</td>
</tr>
<tr>
<td>t</td>
<td>0.090</td>
</tr>
<tr>
<td>u</td>
<td>0.030</td>
</tr>
<tr>
<td>v</td>
<td>0.010</td>
</tr>
<tr>
<td>w</td>
<td>0.015</td>
</tr>
<tr>
<td>x</td>
<td>0.005</td>
</tr>
<tr>
<td>y</td>
<td>0.020</td>
</tr>
<tr>
<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,
  $$\varphi(i) = 0.1 p(6 - i) + 0.1 p(7 - i) + 0.1 p(10 - i) + 0.3 p(14 - i) + 0.2 p(17 - i) + 0.1 p(20 - i) + 0.1 p(25 - i)$$

- $p(x)$ is frequency of character $x$ in English
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></td>
<td></td>
<td>25</td>
<td>0.0430</td>
</tr>
</tbody>
</table>
The Result

• Most probable keys, based on $\varphi$:
  – $i = 6$, $\varphi(i) = 0.0660$
    • plaintext EBIIL TLOLA
  – $i = 10$, $\varphi(i) = 0.0635$
    • plaintext AXEEH PHKEW
  – $i = 3$, $\varphi(i) = 0.0575$
    • plaintext HELLO WORLD
  – $i = 14$, $\varphi(i) = 0.0535$
    • plaintext WTAAD LDGAS

• Only English phrase is for $i = 3$
  – That’s the key (3 or ‘D’)
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
Vigènere 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 VIGVIGVIGVIGVIGVIGVIGVIGV
    - plain THEBOYHASTHEBALL
    - cipher OPKWWECIYOPKWIRG
Relevant Parts of Tableau

- Tableau shown has 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”)

<table>
<thead>
<tr>
<th>A</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>L</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>
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

• Approach
  – Establish period; call it $n$
  – Break message into $n$ parts, each part being enciphered using the same key letter
  – Solve each part
    • You can leverage one part from another

• We will show each step
The Target Cipher

- We want to break this cipher:

  ADQYS MIUSB OXKKT MIBHK IZOOO
  EQOOG IFBAG KAUMF VVTAA CIDTW
  MOCIO EQOOG BMBFV ZGGWP CIEKQ
  HSNEW VECNE DLAAV RWKXS VNSVP
  HCEUT QOIOF MEGJS WTPCH AJMOC
  HIUIX
Establish Period

• Kaskski: *repetitions in the ciphertext occur when characters of the key appear over the same characters in the plaintext*

• Example:

  key    VIGVIGVIGVIGVIGV
  plain  THEBOYHASTHEBALL
  cipher  OPKWWECIYOPKWIRG

Note the key and plaintext line up over the repetitions (underlined). As distance between repetitions is 9, the period is a factor of 9 (that is, 1, 3, or 9)
# Repetitions in Example

<table>
<thead>
<tr>
<th>Letters</th>
<th>Start</th>
<th>End</th>
<th>Distance</th>
<th>Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>MI</td>
<td>5</td>
<td>15</td>
<td>10</td>
<td>2, 5</td>
</tr>
<tr>
<td>OO</td>
<td>22</td>
<td>27</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>OEQQOOG</td>
<td>24</td>
<td>54</td>
<td>30</td>
<td>2, 3, 5</td>
</tr>
<tr>
<td>FV</td>
<td>39</td>
<td>63</td>
<td>24</td>
<td>2, 2, 2, 3</td>
</tr>
<tr>
<td>AA</td>
<td>43</td>
<td>87</td>
<td>44</td>
<td>2, 2, 11</td>
</tr>
<tr>
<td>MOC</td>
<td>50</td>
<td>122</td>
<td>72</td>
<td>2, 2, 2, 3, 3</td>
</tr>
<tr>
<td>QO</td>
<td>56</td>
<td>105</td>
<td>49</td>
<td>7, 7</td>
</tr>
<tr>
<td>PC</td>
<td>69</td>
<td>117</td>
<td>48</td>
<td>2, 2, 2, 2, 3</td>
</tr>
<tr>
<td>NE</td>
<td>77</td>
<td>83</td>
<td>6</td>
<td>2, 3</td>
</tr>
<tr>
<td>SV</td>
<td>94</td>
<td>97</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>CH</td>
<td>118</td>
<td>124</td>
<td>6</td>
<td>2, 3</td>
</tr>
</tbody>
</table>
Estimate of Period

- OEQOOG is probably not a coincidence
  - It’s too long for that
  - Period may be 1, 2, 3, 5, 6, 10, 15, or 30
- Most others (7/10) have 2 in their factors
- Almost as many (6/10) have 3 in their factors
- Begin with period of $2 \times 3 = 6$
Check on Period

• Index of coincidence is probability that two randomly chosen letters from ciphertext will be the same

• Tabulated for different periods:

<table>
<thead>
<tr>
<th>Period</th>
<th>Index of Coincidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.066</td>
</tr>
<tr>
<td>3</td>
<td>0.047</td>
</tr>
<tr>
<td>5</td>
<td>0.044</td>
</tr>
<tr>
<td>2</td>
<td>0.052</td>
</tr>
<tr>
<td>4</td>
<td>0.045</td>
</tr>
<tr>
<td>10</td>
<td>0.041</td>
</tr>
<tr>
<td>Large</td>
<td>0.038</td>
</tr>
</tbody>
</table>
Compute IC

- IC = \left[ n \left( n - 1 \right) \right]^{-1} \sum_{0 \leq i \leq 25} \left[ F_i \left( F_i - 1 \right) \right]
  
  where n is length of ciphertext and F_i the number of times character i occurs in ciphertext

- Here, IC = 0.043
  
  - Indicates a key of slightly more than 5
  
  - A statistical measure, so it can be in error, but it agrees with the previous estimate (which was 6)
Splitting Into Alphabets

alphabet 1: AIKHOIATTOBGEEERNEOSAI
alphabet 2: DUKKEFUAWEMGKWDSUFWJU
alphabet 3: QSTIQBMAMQBWQLKVTMTMI
alphabet 4: YBMZOAFCOOFPHEAXPQEPOX
alphabet 5: SOIOOGVICOVCSVASHOGCC
alphabet 6: MXBOGKVVDIGZINNVVCIJHH

• ICs (#1, 0.069; #2, 0.078; #3, 0.078; #4, 0.056; #5, 0.124; #6, 0.043) indicate all alphabets have period 1, except #4 and #6; assume statistics off
Frequency Examination

<table>
<thead>
<tr>
<th></th>
<th>ABCDEFGHIJKLMNOPQRSTUVWXYZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>31004011301001300112000000</td>
</tr>
<tr>
<td>2</td>
<td>10022210013010000010404000</td>
</tr>
<tr>
<td>3</td>
<td>12000000201140004013021000</td>
</tr>
<tr>
<td>4</td>
<td>21102201000010431000000211</td>
</tr>
<tr>
<td>5</td>
<td>10500021200000500030020000</td>
</tr>
<tr>
<td>6</td>
<td>01110022311012100000030101</td>
</tr>
</tbody>
</table>

Letter frequencies are (H high, M medium, L low):

HMMMMHMHHMMMMMMHMLHHHMLLLLLL
Begin Decryption

- First matches characteristics of unshifted alphabet
- Third matches if I shifted to A
- Sixth matches if V shifted to A
- Substitute into ciphertext (bold are substitutions)

\[
\begin{align*}
\text{ADIYS} & \quad \text{RIUKB} & \quad \text{OCKKL} & \quad \text{MIGHKAZOTO} & \quad \text{EIOOL} \\
\text{IFTAG} & \quad \text{PAUEF} & \quad \text{VATAS} & \quad \text{CIITW} & \quad \text{EOCNO} & \quad \text{EIOOL} \\
\text{BMTFV} & \quad \text{EGGOP} & \quad \text{CNEKI} & \quad \text{HSSEW} & \quad \text{NECSE} & \quad \text{DDAAA} \\
\text{RWCXS} & \quad \text{ANSNP} & \quad \text{HHEUL} & \quad \text{QONOF} & \quad \text{EEGOS} & \quad \text{WLPCM} \\
\text{AJEOC} & \quad \text{MIUAX}
\end{align*}
\]
Look For Clues

- **AJE** in last line suggests “are”, meaning second alphabet maps A into S:
  
  ALIYS RICKB OCKSL MIGHS AZOTO
  MIOOL INTAG PACEF VATIS CIITE
  EOCNO MIOOL BUTFV EGOOP CNESI
  HSSEE NECSE LDAAA RECXS ANANP
  HHECL QONON EEGOS ELPCM AREOC
  MICAX
Next Alphabet

• MICAX in last line suggests “mical” (a common ending for an adjective), meaning fourth alphabet maps O into A:

ALIMS  RICKP  OCKSL  AIGHS  ANOTO  MICOL
INTOG  PACET  VATIS  QITE  ECCNO  MICOL
BUTTV  EGOOD  CNESI  VSSEE  NSCSE  LDOAA
RECLS  ANAND  HHECL  EONON  ESGOS  ELDCM
ARECC  MICAL
Got It!

- QI means that U maps into I, as Q is always followed by U:

  ALIME  RICKP  ACKSL  AUGHS  ANATO  MICAL
  INTOS  PACET  HATIS  QUITE  ECONO  MICAL
  BUTTH  EGOOD  ONESI  VESEE  NSOSE  LDOMA
  RECLE  ANAND  THECL  EANON  ESSOS  ELDOM
  ARECO  MICAL
One-Time Pad

• A Vigenère cipher with a random key at least as long as the message
  – Provably unbreakable
  – Why? Look at ciphertext DXQR. Equally likely to correspond to plaintext DOIT (key AJIY) and to 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
  - basic unit is the bit
  - performs both substitution and transposition (permutation) on the bits

- Cipher consists of 16 rounds (iterations) each with a round key generated from the user-supplied key
Generation of Round Keys

- Round keys are 48 bits each
Encipherment

input

IP

L₀ \rightarrow R₀ \rightarrow ! f \rightarrow K₁

L₁ = R₀ \quad R₁ = \text{L₀} \oplus f(\text{R₀}, K₁)

R₁₆ = \text{L₁₅} \oplus f(\text{R₁₅}, K₁₆)

L₁₆ = \text{R₁₅}

output

IP⁻¹
The $f$ Function

$R_{i-1}$ (32 bits) -> $E$ -> $R_{i-1}$ (48 bits) + $K_i$ (48 bits)

$S1$ $S2$ $S3$ $S4$ $S5$ $S6$ $S7$ $S8$

6 bits into each

$32$ bits

4 bits out of each
Controversy

• Considered too weak
  – Diffie, Hellman said in a few years technology would allow DES to be broken in days
    • Design using 1999 technology published
  – Design decisions not public
    • S-boxes may have backdoors
Undesirable Properties

- 4 weak keys
  - They are their own inverses
- 12 semi-weak keys
  - Each has another semi-weak key as inverse
- Complementation property
  - $\text{DES}_k(m) = c \Rightarrow \text{DES}_k(m') = c'$
- S-boxes exhibit irregular properties
  - Distribution of odd, even numbers non-random
  - Outputs of fourth box depends on input to third box
Differential Cryptanalysis

• A chosen ciphertext attack
  – Requires \(2^{47}\) plaintext, ciphertext pairs

• Revealed several properties
  – Small changes in S-boxes reduce the number of pairs needed
  – Making every bit of the round keys independent does not impede attack

• Linear cryptanalysis improves result
  – Requires \(2^{43}\) plaintext, ciphertext pairs
DES Modes

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

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

- **Encrypt-Decrypt-Encrypt Mode (2 keys: $k$, $k'$)**
  - $c = \text{DES}_k(\text{DES}_{k'}^{-1}(\text{DES}_k(m)))$

- **Encrypt-Encrypt-Encrypt Mode (3 keys: $k$, $k'$, $k''$)**
  - $c = \text{DES}_k(\text{DES}_{k'}(\text{DES}_{k''}(m)))$
CBC Mode Encryption

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

\[
\begin{align*}
\text{init. vector} & \rightarrow c_1 \rightarrow \text{DES} \rightarrow m_1 \\
& \quad \downarrow \oplus \\
& \quad c_2 \rightarrow \text{DES} \rightarrow m_2 \\
& \quad \downarrow \oplus \\
& \quad \ldots
\end{align*}
\]
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
Current Status of DES

• Design for computer system, associated software that could break any DES-enciphered message in a few days published in 1998
• Several challenges to break DES messages solved using distributed computing
• NIST selected Rijndael as Advanced Encryption Standard, successor to DES
  – Designed to withstand attacks that were successful on DES
Public Key Cryptography

• Two keys
  – *Private key* known only to individual
  – *Public key* available to anyone
    • Public key, private key inverses

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

1. It must be computationally easy to encipher or decipher a message given the appropriate key
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