Secret Key Systems (block encoding)

Encrypting a small block of text (say 64 bits)

General considerations for cipher design:
Secret Key Systems (block encoding)

Encrypting a small block of text (say 64 bits)

General considerations for cipher design:
- Encrypted data should look random.
  As though someone flipped a fair coin 64 times and heads means 1 and tails 0.
  Any change in one bit of output corresponds to a huge change in the input (bits are uncorrelated).
  Should be about as many 1's as 0's usually.
Secret Key Systems (block encoding)

Encrypting a small block of text (say 64 bits)

General considerations for cipher design:

- **Encrypted data should look random.**
  As though someone flipped a fair coin 64 times and heads means 1 and tails 0.
  Any change in one bit of output corresponds to a huge change in the input (bits are uncorrelated).
  Should be about as many 1's as 0's usually.

- **Try to spread the influence of each input bit to all output bits**
  Any change in one input bit should have 50% chance of changing any of the output bits (hence many rounds).
Secret Key Systems (block encoding)

Encrypting a small block of text (say 64 bits)

General considerations for cipher design:

- **Encrypted data should look random.**
  As though someone flipped a fair coin 64 times and heads means 1 and tails 0.
  Any change in one bit of output corresponds to a huge change in the input (bits are uncorrelated).
  Should be about as many 1's as 0's usually.

- **Try to spread the influence of each input bit to all output bits**
  Any change in one input bit should have 50% chance of changing any of the output bits (hence many rounds).

- **Operations should be invertible** – hence xor and table lookup.
  Use of one key for both encryption and decryption.
Secret Key Systems (block encoding)

Encrypting a small block of text (say 64 bits)

**General considerations for cipher design:**
- Encrypted data should look random.
  As though someone flipped a fair coin 64 times and heads means 1 and tails 0.
  Any change in one bit of output corresponds to a huge change in the input (bits are uncorrelated).
  Should be about as many 1's as 0's usually.
- Try to spread the influence of each input bit to all output bits
  Any change in one input bit should have 50% chance of changing any of the output bits (hence many rounds).
- Operations should be invertible – hence xor and table lookup.
  Use of one key for both encryption and decryption.
- Attacks may be mitigated if they rely on operations that are not efficiently implemented in hardware yet allow normal operation to complete efficiently, even in software (permute).
Secret Key Systems (block encoding)

Encrypting a small block of text (say 64 bits)

**What is considered a successful attack?**

Suppose some plaintext is known (a crib) and it is desired to find the key.

If the cipher were generating truly random output, an attack on a key should take $2^{55}$ tries, on the average, for 56 bit keys.

If someone can find a way to guarantee finding a key in $2^{50}$ tries even, then the cipher is considered broken.
Secret Key Systems - DES

IBM/NSA 1977 - 64 bit blocks, 56 bit key, 8 bits parity

64 bit input block

32 bit $L_i$

32 bit $R_i$

Mangler Function

$K_i$

Encryption Round

$64$ bit $L_{i+1}$ $32$ bit $R_{i+1}$

64 bit output block
Secret Key Systems - DES

IBM/NSA 1977 - 64 bit blocks, 56 bit key, 8 bits parity

64 bit input block

32 bit $L_i$  \rightarrow  32 bit $R_i$

Mangler Function

32 bit $L_{i+1}$  \rightarrow  32 bit $R_{i+1}$

$\oplus$  $K_i$

64 bit output block

64 bit output block

32 bit $L_i$  \rightarrow  32 bit $R_i$

Mangler Function

32 bit $L_{i+1}$  \rightarrow  32 bit $R_{i+1}$

$\oplus$  $K_i$

64 bit input block

Encryption Round

Decryption Round
Secret Key Systems - DES

Generating per round keys $K_1 \ K_2 \ ... \ K_{16}$ from the 56 bit Key + 8 parity bits

| Key bits: | 1...8 | 9...16 | 17...24 | 25...32 | 33...40 | 41...48 | 49...56 | 57...64 |
Secret Key Systems - DES

Generating per round keys $K_1, K_2, \ldots, K_{16}$ from the 56 bit Key + 8 parity bits

Key bits:

<table>
<thead>
<tr>
<th>1...8</th>
<th>9...16</th>
<th>17...24</th>
<th>25...32</th>
<th>33...40</th>
<th>41...48</th>
<th>49...56</th>
<th>57...64</th>
</tr>
</thead>
</table>

$C_0$:
57 49 41 33 25 17 9 1 58 50 42 34 26 18 10 2 59 51 43 35 27 19 11 3 60 52 44 36

$D_0$:
63 55 47 39 31 23 15 7 62 54 46 38 30 22 14 6 61 53 45 37 29 21 13 5 28 20 12 4
Secret Key Systems - DES

Generating per round keys $K_1, K_2, \ldots, K_{16}$ from the 56 bit Key + 8 parity bits

**Key bits:**

<table>
<thead>
<tr>
<th>1...8</th>
<th>9...16</th>
<th>17...24</th>
<th>25...32</th>
<th>33...40</th>
<th>41...48</th>
<th>49...56</th>
<th>57...64</th>
</tr>
</thead>
</table>

$C_0$:
57 49 41 33 25 17 9 1 58 50 42 34 26 18 10 2 59 51 43 35 27 19 11 3 60 52 44 36

$D_0$:
63 55 47 39 31 23 15 7 62 54 46 38 30 22 14 6 61 53 45 37 29 21 13 5 28 20 12 4

Each round: $K_i$ has 48 bits assembled in 2 halves permuted from 24 bits each of $C_i$ and $D_i$, $K_{i+1}$ is obtained by rotating $C_i$ and $D_i$ left to form $C_{i+1}$ and $D_{i+1}$ (rotation is 1 bit for rounds 1,2,9,16 and 2 bits for other rounds)
Secret Key Systems - DES

Generating per round keys $K_1, K_2 \ldots K_{16}$ from the 56 bit Key + 8 parity bits

Key bits:

<table>
<thead>
<tr>
<th>1...8</th>
<th>9...16</th>
<th>17...24</th>
<th>25...32</th>
<th>33...40</th>
<th>41...48</th>
<th>49...56</th>
<th>57...64</th>
</tr>
</thead>
</table>

$C_0$:  
57 49 41 33 25 17 9 1 58 50 42 34 26 18 10 2 59 51 43 35 27 19 11 3 60 52 44 36

$D_0$:  
63 55 47 39 31 23 15 7 62 54 46 38 30 22 14 6 61 53 45 37 29 21 13 5 28 20 12 4

Each round: $K_i$ has 48 bits assembled in 2 halves permuted from 24 bits each of $C_i$ and $D_i$, $K_{i+1}$ is obtained by rotating $C_i$ and $D_i$ left to form $C_{i+1}$ and $D_{i+1}$ (rotation is 1 bit for rounds 1,2,9,16 and 2 bits for other rounds)

Permutations:

Left half $C_i$: (9,18,22,25 are missing)  
14 17 11 24 1 5 3 28 15 6 21 10 23 19 12 4 26 8 16 7 27 20 13 2

Right half $D_i$: (35,38,43,54 are missing)  
41 52 31 37 47 55 30 40 51 45 33 48 44 49 39 56 34 53 46 42 50 36 29 32
Secret Key Systems - DES

The Mangler function: mixes 32 bit input with 48 bit key to produce 32 bits

\[ 32 \text{ bit } R_i \]

\[ \oplus \]

\[ K_i \]

\[ 32 \text{ bit } R_{i+1} \]
Secret Key Systems - DES

The Mangler function: mixes 32 bit input with 48 bit key to produce 32 bits

1. Expansion of input bits:

\[
32 \text{ bit } R_i \xrightarrow{\oplus} \text{Mangler Function} \xrightarrow{K_i} 32 \text{ bit } R_{i+1}
\]
Secret Key Systems - DES

The Mangler function: mixes 32 bit input with 48 bit key to produce 32 bits

1. Expansion of input bits:
Secret Key Systems - DES

The Mangler function: mixes 32 bit input with 48 bit key to produce 32 bits

1. Expansion of input bits:

32 bit $R_i$

Mangler Function

32 bit $R_{i+1}$

K

$K_i$ (48 bits)

Expanded input (48 bits)

$R_i$ (32 bits)
Secret Key Systems - DES

The Mangler function: mixes 32 bit input with 48 bit key to produce 32 bits

1. Expansion of input bits:

   32 bit \( R_i \)

   Mangler Function

   32 bit \( R_{i+1} \)

   \( K_i \)

   \( R_i \) (32 bits) 

   \( K_i \) (48 bits)

   Expanded input (48 bits)

2. Mixing with key:

   S-Box \( S_{Box_i} \)

   4 bits 4 bits 4 bits 4 bits 4 bits 4 bits 4 bits 4 bits

   \( \oplus \)
Secret Key Systems - DES

The Mangler function: mixes 32 bit input with 48 bit key to produce 32 bits

1. Expansion of input bits:

2. Mixing with key:
Secret Key Systems - DES

The **S-Box**: maps 6 bit blocks to 4 bit sections

**S-Box**₁ (first 6 bits):

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>1110</td>
<td>0100</td>
<td>1101</td>
<td>0001</td>
<td>0010</td>
<td>1111</td>
</tr>
<tr>
<td>01</td>
<td>0000</td>
<td>1111</td>
<td>0111</td>
<td>0100</td>
<td>1110</td>
<td>0010</td>
</tr>
<tr>
<td>10</td>
<td>0100</td>
<td>0001</td>
<td>1110</td>
<td>1000</td>
<td>1101</td>
<td>0110</td>
</tr>
<tr>
<td>11</td>
<td>1111</td>
<td>1100</td>
<td>1000</td>
<td>0010</td>
<td>0100</td>
<td>1001</td>
</tr>
</tbody>
</table>

Input bits 1 and 6

Input bits 2,3,4,5

**Final permutation:**

16 7 20 21 29 12 28 17 1 15 23 26 5 18 31 10 2 8 24 14 32 27 3 9 19 13 30 6 22 11 4 25
Weak and semi-weak keys:
If key is such that $C_0$ or $D_0$ are:

1) alternating 'F' and 'E': \(0xFEFEFE\ldots\)
2) \(0xE0E0E0F1F1F1\), \(0xF1F1F1F1E0E0E0\)
3) alternating 1s and 0s: \(0x010101\ldots\)

weak: \(E(E(M)) = M\)
Secret Key Systems - DES

Weak and semi-weak keys:

If key is such that $C_0$ or $D_0$ are:

1) alternating 'F' and 'E': 0xFEFEFEFE...
2) 0xE0E0E0E0F1F1F1F1, 0xF1F1F1F1E0E0E0E0
3) alternating 1s and 0s: 0x010101...

weak: $E(E(M)) = M$

0x011F011F010E010E and 0x1F011F010E010E01
0x01E001E001F101F1 and 0xE001E001F101F101
0X01FE01FE01FE01FE and 0xFE01FE01FE01FE01

semi-weak: $E_{K_1} (E_{K_2} (M)) = M$ (six total pairs)
Secret Key Systems - DES

Weak and semi-weak keys:
If key is such that $C_0$ or $D_0$ are:

1) alternating 'F' and 'E': 0xFEFEFE...
2) 0xE0E0E0F1F1F1F1, 0xF1F1F1F1E0E0E0E0
3) alternating 1s and 0s: 0x010101...

weak: $E(E(M)) = M$
0x011F011F010E010E and 0x1F011F010E010E01
0x01E001E001F101F1 and 0xE001E001F101F101
0x01FE01FE01FE01FE and 0xFE01FE01FE01FE01

semi-weak: $E_{K_1}(E_{K_2}(M)) = M$ (six total pairs)

Discussion:
1. Designed to be resistant to differential attacks – this type of attack was kept secret by NSA/IBM because other current cryptosystems were vulnerable
2. Changing S-Boxes has resulted in provably weaker system
Secret Key Systems - 3DES

Two keys $K_1$ and $K_2$:

\[
\begin{align*}
\begin{array}{c}
K_1 \ K_2 \ K_1 \\
\downarrow \quad \downarrow \quad \downarrow \\
m \rightarrow E \rightarrow D \rightarrow E \rightarrow c
\end{array}
\end{align*}
\]

encryption

\[
\begin{align*}
\begin{array}{c}
K_1 \ K_2 \ K_1 \\
\downarrow \quad \downarrow \quad \downarrow \\
c \rightarrow D \rightarrow E \rightarrow D \rightarrow m
\end{array}
\end{align*}
\]

decryption
Secret Key Systems - 3DES

Two keys $K_1$ and $K_2$:

$$
\begin{array}{c}
  \downarrow & \downarrow & \downarrow \\
  m & \rightarrow & E \rightarrow D \rightarrow E \rightarrow c \\
\end{array}
$$

encryption

$$
\begin{array}{c}
  \downarrow & \downarrow & \downarrow \\
  c & \rightarrow & D \rightarrow E \rightarrow D \rightarrow m \\
\end{array}
$$

decryption

Why not 2DES

1. Double encryption with the same key still requires searching $2^{56}$ keys
Secret Key Systems - 3DES

Two keys $K_1$ and $K_2$:

$m \rightarrow E \rightarrow D \rightarrow E \rightarrow c \quad c \rightarrow D \rightarrow E \rightarrow D \rightarrow m$

encryption \hspace{2cm} decryption

Why not 2DES

1. Double encryption with the same key still requires searching $2^{56}$ keys
2. Double encryption with two different keys is just as vulnerable as DES due to the following, assuming some $<m,c>$ pairs are known:
Secret Key Systems - 3DES

Two keys $K_1$ and $K_2$:

\[
\begin{array}{cccc}
K_1 & K_2 & K_1 \\
\downarrow & \downarrow & \downarrow \\
m & \rightarrow & E \rightarrow D \rightarrow E & \rightarrow c
\end{array}
\]

encryption

\[
\begin{array}{cccc}
K_1 & K_2 & K_1 \\
\downarrow & \downarrow & \downarrow \\
c & \rightarrow & D \rightarrow E \rightarrow D & \rightarrow m
\end{array}
\]

decryption

Why not 2DES

1. Double encryption with the same key still requires searching $2^{56}$ keys
2. Double encryption with two different keys is just as vulnerable as DES due to the following, assuming some $\langle m, c \rangle$ pairs are known:

\[
\begin{array}{c|c|c}
m: & K_1 & E(K_1, m) \\
\hline
101010 & 1000011 \\
\vdots & \vdots & \vdots \\
100010 & 0101111 \\
001011 & 0001101 \\
\end{array}
\]

$2^{56}$
Secret Key Systems - 3DES

Two keys $K_1$ and $K_2$:

\[
\begin{array}{ccc}
K_1 & K_2 & K_1 \\
\downarrow & \downarrow & \downarrow \\
m \rightarrow E \rightarrow D \rightarrow E \rightarrow c \\
\end{array}
\]

\[
\begin{array}{ccc}
K_1 & K_2 & K_1 \\
\downarrow & \downarrow & \downarrow \\
c \rightarrow D \rightarrow E \rightarrow D \rightarrow m \\
\end{array}
\]

encryption
decryption

Why not 2DES

1. Double encryption with the same key still requires searching $2^{56}$ keys
2. Double encryption with two different keys is just as vulnerable as DES due to the following, assuming some $<m,c>$ pairs are known:

\[
\begin{array}{c|c|c}
2^{56} & m: K_1 & E(K_1, m) \\
\hline
101010 & 1000011 \\
\vdots & \vdots \\
100010 & 0101111 \\
001011 & 0001101 \\
\end{array}
\]

\[
\begin{array}{c|c|c}
2^{56} & c: K_2 & D(K_2, c) \\
\hline
101110 & 0001101 \\
\vdots & \vdots \\
001110 & 1000011 \\
001011 & 0001101 \\
\end{array}
\]
Secret Key Systems - 3DES

Two keys $K_1$ and $K_2$:

\[
\begin{array}{ccc}
K_1 & K_2 & K_1 \\
\downarrow & \downarrow & \downarrow \\
m & \rightarrow E & \rightarrow D & \rightarrow E & \rightarrow c \\
\end{array}
\]

encryption

\[
\begin{array}{ccc}
K_1 & K_2 & K_1 \\
\downarrow & \downarrow & \downarrow \\
c & \rightarrow D & \rightarrow E & \rightarrow D & \rightarrow m \\
\end{array}
\]

decryption

Why not 2DES

1. Double encryption with the same key still requires searching $2^{56}$ keys
2. Double encryption with two different keys is just as vulnerable as DES due to the following, assuming some $<m,c>$ pairs are known:

\[
\begin{array}{lll}
m: & K_1 & E(K_1,m) \\
\hline
101010 & 1000011 \\
& ... & ... \\
100010 & 0101111 \\
001011 & 0001101 \\
2^{56} & \{ & \\
\hline
\end{array}
\]

\[
\begin{array}{lll}
c: & K_2 & D(K_2,c) \\
\hline
101110 & 0001101 \\
& ... & ... \\
001110 & 1000011 \\
001011 & 0001101 \\
2^{56} & \{ & \\
\hline
\end{array}
\]
Secret Key Systems - 3DES

Two keys $K_1$ and $K_2$:

$K_1 \quad K_2 \quad K_1$

$m \rightarrow E \rightarrow D \rightarrow E \rightarrow c$

encryption

$K_1 \quad K_2 \quad K_1$

$c \rightarrow D \rightarrow E \rightarrow D \rightarrow m$

decryption

Why not 2DES

1. Double encryption with the same key still requires searching $2^{56}$ keys
2. Double encryption with two different keys is just as vulnerable as DES
due to the following, assuming some $<m,c>$ pairs are known:

<table>
<thead>
<tr>
<th>$m$</th>
<th>$K_1$</th>
<th>$E(K_1,m)$</th>
<th>$c$</th>
<th>$K_2$</th>
<th>$D(K_2,c)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>101010</td>
<td>101110</td>
<td>001011</td>
<td>1000011</td>
<td>0001101</td>
<td></td>
</tr>
<tr>
<td>100010</td>
<td>0101111</td>
<td>001110</td>
<td>0101111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>001011</td>
<td>0001101</td>
<td>001110</td>
<td>1000011</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Test matches on other $<m,c>$ pairs

$2^{56}$ $2^{56}$ $m \rightarrow E(K_1,m)$ $D(K_2,c) \leftarrow c$
Secret Key Systems - 3DES

Why not 2DES

3. Double encryption with two different keys is just as vulnerable as DES due to the following, assuming some \(<m,c>\) pairs are known:

How many \(<m,c>\) pairs do you need?

- \(2^{64}\) possible blocks
- \(2^{56}\) table entries
- each block has probability \(2^{-8} = 1/256\) of showing in a table
- probability a block is in both tables is \(2^{-16}\)
- average number of matches is \(2^{-16} \times 2^{64} = 2^{48}\)
- average number of matches for two \(<m,c>\) pairs is about \(2^{32}\)
- for three \(<m,c>\) pairs is about \(2^{16}\)
- for four \(<m,c>\) pairs is about \(2^{0}\)

4. Triple encryption with two different keys
   - 112 bits of key is considered enough
   - No added security with three keys, encrypting three times
Secret Key Systems - 3DES

CBC with 3DES:

\[ m_i \oplus E(K_1 \oplus D(K_2 \oplus E(K_1))) = c_i \]

\[ m_{i+1} \oplus E(K_1 \oplus D(K_2 \oplus E(K_1))) = c_{i+1} \]
Secret Key Systems - 3DES

CBC with 3DES:
1. On the outside – same attack as with CBC – change a block with side effect of garbling another
2. On the inside – attempt at changing a block results in all blocks garbled to the end of the message.
3. On the inside – use three times as much hardware to pipeline encryptions resulting in DES speeds.
4. On the outside – EDE simply is a drop-in replacement for what might have been there before.
Secret Key Systems - IDEA

ETH Zuria 1991 - 64 bit blocks, 128 bit key, 8.5 rounds

Odd Encryption Round

- Multiplication mod $2^{16} + 1$. 0 is treated as $2^{16}$ (invertible)
- Addition but throw away carries
Secret Key Systems - IDEA

ETH Zuria 1991 - 64 bit blocks, 128 bit key, 8.5 rounds

Even Encryption Round

\[
\begin{align*}
Y_{in} &= X_{i,a} \otimes X_{i,b} \\
Z_{in} &= X_{i,c} \otimes X_{i,d} \\
Y_{out} &= ((K_{i,e} \otimes Y_{in}) \circ Z_{in} \otimes K_{i,f}) \\
Z_{out} &= (K_{i,e} \otimes Y_{in}) \circ Y_{out} \\
X_{i+1,a} &= X_{i,a} \circ Y_{out} \\
X_{i+1,b} &= X_{i,b} \circ Y_{out} \\
X_{i+1,c} &= X_{i,c} \circ Z_{out} \\
X_{i+1,d} &= X_{i,d} \circ Z_{out}
\end{align*}
\]
Secret Key Systems - IDEA

Key generation  52 16 bit keys needed (2 per even round 4 per odd):

128 bit key

\[K_{1,a} \quad K_{1,b} \quad K_{1,c} \quad K_{1,d} \quad K_{2,e} \quad K_{2,f} \quad K_{3,a} \quad K_{3,b}\]
Secret Key Systems - IDEA

Key generation  52 16 bit keys needed (2 per even round 4 per odd):

128 bit key

$K_{1,a}$  $K_{1,b}$  $K_{1,c}$  $K_{1,d}$  $K_{2,e}$  $K_{2,f}$  $K_{3,a}$  $K_{3,b}$

128 bit key

$K_{3,c}$  $K_{3,d}$  $K_{4,e}$  $K_{4,f}$  $K_{5,a}$  $K_{5,b}$  $K_{5,c}$  $K_{5,d}$

25 bits
Secret Key Systems - IDEA

IDEA has been proven to be successful against many cipher attack methods.

It appears to have no algebraic or linear type of weakness

(linear: develop equations relating plaintext, ciphertext, key bits whose probability of holding over the space of all possible values of their variables is close to 1 or 0 – then use these equations to get key bits from known cipher/text-plaintext pairs)

Immune to differential attack under some circumstances

IDEA was 'broken' in 2011 using a meet-in-the-middle attack.
IDEA was 'broken' in 2012 using a narrow-bicliques attack (variant of meet-in-the-middle attack) reducing cryptographic strength by 2 bits.

Patents limited use until 2012 when all expired.

There exist some weak keys but they are rare.
Secure Shell uses the following ciphers for encryption and authentication:

<table>
<thead>
<tr>
<th>Cipher</th>
<th>SSH1</th>
<th>SSH2</th>
</tr>
</thead>
<tbody>
<tr>
<td>DES</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>3DES</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>IDEA</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Blowfish</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Twofish</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Arcfour</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>AES</td>
<td></td>
<td>yes</td>
</tr>
<tr>
<td>RSA</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>DSA</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>
Secret Key Systems - AES

NIST (2001) parameterized key size (128 bits to 256 bits)

4\(N_b\) octet input

4\(N_k\) octet key

\[
\begin{array}{cccc}
a_{0,0} & a_{0,3} \\
a_{1,0} & a_{1,3} \\
a_{2,0} & a_{2,3} \\
a_{3,0} & a_{3,3} \\
\end{array}
\]

key expansion

\[
\begin{array}{cccc}
k_{0,0} & k_{0,3} \\
k_{1,0} & k_{1,3} \\
k_{2,0} & k_{2,3} \\
k_{3,0} & k_{3,3} \\
\end{array}
\]

\(K_0\)

\(K_1\)

round 1
**Secret Key Systems - AES**

**The State:** An array of four rows and $N_b$ columns – each element is a byte.

**Initially:** next block of $4N_b$ input bytes.

**Execution:** all operations are performed on the State.

**Example:** $N_b = 4$

<table>
<thead>
<tr>
<th>$in_0$</th>
<th>$in_4$</th>
<th>$in_8$</th>
<th>$in_{12}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$in_1$</td>
<td>$in_5$</td>
<td>$in_9$</td>
<td>$in_{13}$</td>
</tr>
<tr>
<td>$in_2$</td>
<td>$in_6$</td>
<td>$in_{10}$</td>
<td>$in_{14}$</td>
</tr>
<tr>
<td>$in_3$</td>
<td>$in_7$</td>
<td>$in_{11}$</td>
<td>$in_{15}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$s_{0,0}$</th>
<th>$s_{0,1}$</th>
<th>$s_{0,2}$</th>
<th>$s_{0,3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_{1,0}$</td>
<td>$s_{1,1}$</td>
<td>$s_{1,2}$</td>
<td>$s_{1,3}$</td>
</tr>
<tr>
<td>$s_{2,0}$</td>
<td>$s_{2,1}$</td>
<td>$s_{2,2}$</td>
<td>$s_{2,3}$</td>
</tr>
<tr>
<td>$s_{3,0}$</td>
<td>$s_{3,1}$</td>
<td>$s_{3,2}$</td>
<td>$s_{3,3}$</td>
</tr>
</tbody>
</table>
Secret Key Systems - AES

**Addition**: modulo 2 addition (xor) of polynomials of maximum degree 7

**Examples**:

\[(x^6 + x^4 + x^2 + x^1 + 1) + (x^7 + x^1 + 1) = x^7 + x^6 + x^4 + x^2\]  (polynomial)

01010111 ⊕ 10000011 = 11010100  (binary notation)

0x57 ⊕ 0x83 = D4  (hexadecimal)
Secret Key Systems - AES

Multiplication of two degree 7 polynomials (bytes):

Just like ordinary multiplication except mod \( m(x) = (x^8 + x^4 + x^3 + x^1 + 1) \)

**Reason**: for each byte there will be an inverse: \( a \times a^{-1} = 1 \mod m(x) \)

**Basis**: \( x \times b = b_7 x^8 + b_6 x^7 + b_5 x^6 + b_4 x^5 + b_3 x^4 + b_2 x^3 + b_1 x^2 + b_0 x \)

Shift \( b \) left by 1, if result has a degree 8 bit, xor with \( m(x) \)

This operation is called \( \text{xtime}(x) = (x << 1) \oplus (((x >> 7) \& 1) \times 0x11b) \)
Secret Key Systems - AES

Multiplication of two degree 7 polynomials (bytes):

Just like ordinary multiplication except mod \( m(x) = (x^8+x^4+x^3+x^1+1) \)

**Reason:** for each byte there will be an inverse: \( a \times a^{-1} = 1 \mod m(x) \)

**Basis:** \( x \times b = b_7x^8 + b_6x^7 + b_5x^6 + b_4x^5 + b_3x^4 + b_2x^3 + b_1x^2 + b_0x \)

Shift \( b \) left by 1, if result has a degree 8 bit, xor with \( m(x) \)

This operation is called \( \text{xtime}(x) = (x << 1) \oplus (((x >> 7) \& 1) \ast 0x11b) \)

**Example:**

\[
\text{xtime}(x^7 + x^5 + x^3 + x^2 + x) = x^6 + x^2 + x + 1 \quad \text{or} \quad 101011100 \oplus 100011011 = 01000111 \quad \text{or} \quad \text{xtime}(0xAE) = 0x47
\]

**Example:**

\[
\begin{align*}
(x^6 + x^4 + x^2 + x^1 + 1) \otimes (x^4 + x + 1) &= 0x57 \otimes 0x13 \\
(x^6 + x^4 + x^2 + x^1 + 1) \otimes x &= \text{xtime}(0x57) = 0xAE \\
(x^6 + x^4 + x^2 + x^1 + 1) \otimes x^2 &= \text{xtime}(0xAE) = 0x47 \\
(x^6 + x^4 + x^2 + x^1 + 1) \otimes x^3 &= \text{xtime}(0x47) = 0x8E \\
(x^6 + x^4 + x^2 + x^1 + 1) \otimes x^4 &= \text{xtime}(0x8E) = 0x7
\end{align*}
\]

\[
0x57 \otimes 0x13 = 0x7 \oplus 0xAE \oplus 0x57 = 0xFE
\]
Secret Key Systems - AES

Byte substitutions (S-Box):

Find the inverse of a polynomial mod \( m(x) \):
\[
a(x) \otimes b(x) \oplus m(x) \otimes c(x) = 1
\]

Example:

\( 0x57 \otimes 0xBF = 1 \) so 0xBF is the inverse of 0x57

S-Box number:

Apply the transformation on the right to the inverse

Example:

\( \text{getSbox}(0x57) = 0x5B \)
Secret Key Systems - AES

The S-Box:

<table>
<thead>
<tr>
<th>x</th>
<th>0 1 2 3 4 5 6 7 8 9 a b c d e f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63 7c 77 7b f2 6b 6f c5 30 01 67 2b fe d7 ab 76</td>
</tr>
<tr>
<td>1</td>
<td>ca 82 c9 7d fa 59 47 f0 ad d4 a2 af 9c a4 72 c0</td>
</tr>
<tr>
<td>2</td>
<td>b7 fd 93 26 36 3f f7 cc 34 a5 e5 f1 71 d8 31 15</td>
</tr>
<tr>
<td>3</td>
<td>04 c7 23 c3 18 96 05 9a 07 12 80 e2 eb 27 b2 75</td>
</tr>
<tr>
<td>4</td>
<td>09 83 2c 1a 1b 6e 5a a0 52 3b d6 b3 29 e3 2f 84</td>
</tr>
<tr>
<td>5</td>
<td>53 d1 00 ed 20 fc b1 5b 6a cb be 39 4a 4c 58 cf</td>
</tr>
<tr>
<td>6</td>
<td>d0 ef aa fb 43 4d 33 85 45 f9 02 7f 50 3c 9f a8</td>
</tr>
<tr>
<td>7</td>
<td>51 a3 40 8f 92 9d 38 f5 bc b6 da 21 10 ff f3 d2</td>
</tr>
<tr>
<td>8</td>
<td>cd 0c 13 ec 5f 97 44 17 c4 a7 7e 3d 64 5d 19 73</td>
</tr>
<tr>
<td>9</td>
<td>60 81 4f dc 22 2a 90 88 46 ee b8 14 de 5e 0b db</td>
</tr>
<tr>
<td>a</td>
<td>e0 32 3a 0a 49 06 24 5c c2 d3 ac 62 91 95 e4 79</td>
</tr>
<tr>
<td>b</td>
<td>e7 c8 37 6d 8d d5 4e a9 6c 56 f4 ea 65 7a ae 08</td>
</tr>
<tr>
<td>c</td>
<td>ba 78 25 2e 1c a6 b4 c6 e8 dd 74 1f 4b bd 8b 8a</td>
</tr>
<tr>
<td>d</td>
<td>70 3e b5 66 48 03 f6 0e 61 35 57 b9 86 c1 1d 9e</td>
</tr>
<tr>
<td>e</td>
<td>e1 f8 98 11 69 d9 8e 94 9b 1e 87 e9 ce 55 28 df</td>
</tr>
<tr>
<td>f</td>
<td>8c a1 89 0d bf e6 42 68 41 99 2d 0f b0 54 bb 16</td>
</tr>
</tbody>
</table>
The Inverse S-Box:

<table>
<thead>
<tr>
<th>x</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>52</td>
<td>09</td>
<td>6a</td>
<td>d5</td>
<td>30</td>
<td>36</td>
<td>a5</td>
<td>38</td>
<td>bf</td>
<td>40</td>
<td>a3</td>
<td>9e</td>
<td>81</td>
<td>f3</td>
<td>d7</td>
<td>fb</td>
</tr>
<tr>
<td>1</td>
<td>7c</td>
<td>e3</td>
<td>39</td>
<td>82</td>
<td>9b</td>
<td>2f</td>
<td>ff</td>
<td>87</td>
<td>34</td>
<td>8e</td>
<td>43</td>
<td>44</td>
<td>c4</td>
<td>de</td>
<td>e9</td>
<td>cb</td>
</tr>
<tr>
<td>2</td>
<td>54</td>
<td>7b</td>
<td>94</td>
<td>32</td>
<td>a6</td>
<td>c2</td>
<td>23</td>
<td>3d</td>
<td>ee</td>
<td>4c</td>
<td>95</td>
<td>0b</td>
<td>42</td>
<td>fa</td>
<td>c3</td>
<td>4e</td>
</tr>
<tr>
<td>3</td>
<td>08</td>
<td>2e</td>
<td>a1</td>
<td>66</td>
<td>28</td>
<td>d9</td>
<td>24</td>
<td>b2</td>
<td>76</td>
<td>5b</td>
<td>a2</td>
<td>49</td>
<td>6d</td>
<td>8b</td>
<td>d1</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>72</td>
<td>f8</td>
<td>f6</td>
<td>64</td>
<td>86</td>
<td>68</td>
<td>98</td>
<td>16</td>
<td>d4</td>
<td>a4</td>
<td>5c</td>
<td>cc</td>
<td>5d</td>
<td>65</td>
<td>b6</td>
<td>92</td>
</tr>
<tr>
<td>5</td>
<td>6c</td>
<td>70</td>
<td>48</td>
<td>50</td>
<td>fd</td>
<td>ed</td>
<td>b9</td>
<td>da</td>
<td>5e</td>
<td>15</td>
<td>46</td>
<td>57</td>
<td>a7</td>
<td>8d</td>
<td>9d</td>
<td>84</td>
</tr>
<tr>
<td>6</td>
<td>90</td>
<td>d8</td>
<td>ab</td>
<td>00</td>
<td>8c</td>
<td>bc</td>
<td>d3</td>
<td>0a</td>
<td>f7</td>
<td>e4</td>
<td>58</td>
<td>05</td>
<td>b8</td>
<td>b3</td>
<td>45</td>
<td>06</td>
</tr>
<tr>
<td>7</td>
<td>d0</td>
<td>2c</td>
<td>1e</td>
<td>8f</td>
<td>ca</td>
<td>3f</td>
<td>0f</td>
<td>02</td>
<td>c1</td>
<td>af</td>
<td>bd</td>
<td>03</td>
<td>01</td>
<td>13</td>
<td>8a</td>
<td>6b</td>
</tr>
<tr>
<td>8</td>
<td>3a</td>
<td>91</td>
<td>11</td>
<td>41</td>
<td>4f</td>
<td>67</td>
<td>dc</td>
<td>ea</td>
<td>97</td>
<td>f2</td>
<td>cf</td>
<td>ce</td>
<td>f0</td>
<td>b4</td>
<td>e6</td>
<td>73</td>
</tr>
<tr>
<td>9</td>
<td>96</td>
<td>ac</td>
<td>74</td>
<td>22</td>
<td>e7</td>
<td>ad</td>
<td>35</td>
<td>85</td>
<td>e2</td>
<td>f9</td>
<td>37</td>
<td>e8</td>
<td>1c</td>
<td>75</td>
<td>df</td>
<td>6e</td>
</tr>
<tr>
<td>a</td>
<td>47</td>
<td>f1</td>
<td>1a</td>
<td>71</td>
<td>1d</td>
<td>29</td>
<td>c5</td>
<td>89</td>
<td>6f</td>
<td>b7</td>
<td>62</td>
<td>0e</td>
<td>aa</td>
<td>18</td>
<td>be</td>
<td>1b</td>
</tr>
<tr>
<td>b</td>
<td>fc</td>
<td>56</td>
<td>3e</td>
<td>4b</td>
<td>c6</td>
<td>d2</td>
<td>79</td>
<td>20</td>
<td>9a</td>
<td>db</td>
<td>c0</td>
<td>fe</td>
<td>78</td>
<td>cd</td>
<td>5a</td>
<td>f4</td>
</tr>
<tr>
<td>c</td>
<td>1f</td>
<td>dd</td>
<td>a8</td>
<td>33</td>
<td>88</td>
<td>07</td>
<td>c7</td>
<td>31</td>
<td>b1</td>
<td>12</td>
<td>10</td>
<td>59</td>
<td>27</td>
<td>80</td>
<td>ec</td>
<td>5f</td>
</tr>
<tr>
<td>d</td>
<td>60</td>
<td>51</td>
<td>7f</td>
<td>a9</td>
<td>19</td>
<td>b5</td>
<td>4a</td>
<td>0d</td>
<td>2d</td>
<td>e5</td>
<td>7a</td>
<td>9f</td>
<td>93</td>
<td>c9</td>
<td>9c</td>
<td>ef</td>
</tr>
<tr>
<td>e</td>
<td>a0</td>
<td>e0</td>
<td>3b</td>
<td>4d</td>
<td>ae</td>
<td>2a</td>
<td>f5</td>
<td>b0</td>
<td>c8</td>
<td>eb</td>
<td>bb</td>
<td>3c</td>
<td>83</td>
<td>53</td>
<td>99</td>
<td>61</td>
</tr>
<tr>
<td>f</td>
<td>17</td>
<td>2b</td>
<td>04</td>
<td>7e</td>
<td>ba</td>
<td>77</td>
<td>d6</td>
<td>26</td>
<td>e1</td>
<td>69</td>
<td>14</td>
<td>63</td>
<td>55</td>
<td>21</td>
<td>0c</td>
<td>7d</td>
</tr>
</tbody>
</table>
Four term polynomials with 8 bit coefficients:
State columns will be represented by such polynomials. We will want to transform columns by multiplication, modulo a polynomial, as was done to get the S-Box. However, in this case the maximum degree is 3 and the coefficients are bytes, not bits. The idea is to multiply a column polynomial by the fixed polynomial $a(x) = 3x^3 + x^2 + x + 2 \mod x^4 + 1$

Equivalently: $a(x) \otimes b(x) = d(x) = d_3x^3 + d_2x^2 + d_1x + d_0$

where $d_0 = (2 \otimes b_0) \oplus (3 \otimes b_1) \oplus (1 \otimes b_2) \oplus (1 \otimes b_3)$
$d_1 = (1 \otimes b_0) \oplus (2 \otimes b_1) \oplus (3 \otimes b_2) \oplus (1 \otimes b_3)$
$d_2 = (1 \otimes b_0) \oplus (1 \otimes b_1) \oplus (2 \otimes b_2) \oplus (3 \otimes b_3)$
$d_3 = (3 \otimes b_0) \oplus (1 \otimes b_1) \oplus (1 \otimes b_2) \oplus (2 \otimes b_3)$

Observe: $d_i$ is influenced by all four bytes of original column

Example: $a(x) \otimes (0xBx^3 + 0xDDx^2 + 0x9x + 0xEE) = 1$
void Cipher () {       // Nr is the number of rounds
  int i, j, round=0;

  // Copy the input PlainText to state array.
  state = in;

  AddRoundKey(0);

  for (round=1 ; round < Nr ; round++) {
    SubBytes();
    ShiftRows();
    MixColumns();
    AddRoundKey(round);
  }

  SubBytes();
  ShiftRows();
  AddRoundKey(Nr);

  // Copy the state array to the Output array.
  out = state;
}
Secret Key Systems - AES

SubBytes ():

<table>
<thead>
<tr>
<th>$s_{0,0}$</th>
<th>$s_{0,1}$</th>
<th>$s_{0,2}$</th>
<th>$s_{0,3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_{1,0}$</td>
<td>$s_{1,1}$</td>
<td>$s_{1,2}$</td>
<td>$s_{1,3}$</td>
</tr>
<tr>
<td>$s_{2,0}$</td>
<td>$s_{2,1}$</td>
<td>$s_{2,2}$</td>
<td>$s_{2,3}$</td>
</tr>
<tr>
<td>$s_{3,0}$</td>
<td>$s_{3,1}$</td>
<td>$s_{3,2}$</td>
<td>$s_{3,3}$</td>
</tr>
</tbody>
</table>

$gSx(s_{0,0})$ $gSx(s_{0,1})$ $gSx(s_{0,2})$ $gSx(s_{0,3})$

$gSx(s_{1,0})$ $gSx(s_{1,1})$ $gSx(s_{1,2})$ $gSx(s_{1,3})$

$gSx(s_{2,0})$ $gSx(s_{2,1})$ $gSx(s_{2,2})$ $gSx(s_{2,3})$

$gSx(s_{3,0})$ $gSx(s_{3,1})$ $gSx(s_{3,2})$ $gSx(s_{3,3})$

Function $gSx(a)$ maps $a$ to the character it indexes in the S-Box
Secret Key Systems - AES

**ShiftRows:**

\[
\begin{array}{cccc}
S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\
S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\
S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\
S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \\
\end{array}
\]

\[
\begin{array}{cccc}
S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\
S_{1,1} & S_{1,2} & S_{1,3} & S_{1,0} \\
S_{2,2} & S_{2,3} & S_{2,0} & S_{2,1} \\
S_{3,3} & S_{3,0} & S_{3,1} & S_{3,2} \\
\end{array}
\]

\[
\begin{array}{|c|c|c|c|}
\hline
N_b & Row & 1 & 2 & 3 \\
\hline
4 & 1 & 2 & 3 \\
6 & 1 & 2 & 3 \\
8 & 1 & 3 & 4 \\
\hline
\end{array}
\]
Secret Key Systems - AES

MixColumns ( ) :

\[
\begin{bmatrix}
    s'_{0,c} \\
    s'_{1,c} \\
    s'_{2,c} \\
    s'_{3,c}
\end{bmatrix} =
\begin{bmatrix}
    2 & 3 & 1 & 1 \\
    1 & 2 & 3 & 1 \\
    1 & 1 & 2 & 3 \\
    3 & 1 & 1 & 2
\end{bmatrix}
\begin{bmatrix}
    s_{0,c} \\
    s_{1,c} \\
    s_{2,c} \\
    s_{3,c}
\end{bmatrix}
\]

Replace state columns by matrix multiplication (⊗ and ⊕) above

Columns are considered as polynomials over GF(2^8) and multiplied mod \( x^4 + 1 \) by a fixed polynomial given by

\[3x^3 + x^2 + x + 2\]

\[\begin{array}{c}
\text{Example:} \\
\begin{bmatrix}
    0xD4 \\
    0xBF \\
    0x5D \\
    0x30
\end{bmatrix} \rightarrow \\
\begin{bmatrix}
    0x04 \\
    0x66 \\
    0x81 \\
    0xEE
\end{bmatrix}
\end{array}\]
Secret Key Systems - AES

MixColumns ():

0x2B 0xD4 0xDE 0xAD
0x41 0xAD 0xAD 0xEC
0xA7 0xDE 0xDE 0x79
0xB3 0xD4 0xD4 0x67
0x56 0x42 0x4C 0xB4
0x67 0x36 0x36 0x36

0x41 0xD4 0xDE 0xAD
0xAD 0xAD 0xEC
0xA7 0xDE 0xDE 0x79
0xB3 0xD4 0xD4 0x67
0x42 0x4C 0xB4 0x36

lookups
Secret Key Systems - AES

AddRoundKey():

<table>
<thead>
<tr>
<th>$s_{0,0}$</th>
<th>$s_{0,1}$</th>
<th>$s_{0,2}$</th>
<th>$s_{0,3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_{1,0}$</td>
<td>$s_{1,1}$</td>
<td>$s_{1,2}$</td>
<td>$s_{1,3}$</td>
</tr>
<tr>
<td>$s_{2,0}$</td>
<td>$s_{2,1}$</td>
<td>$s_{2,2}$</td>
<td>$s_{2,3}$</td>
</tr>
<tr>
<td>$s_{3,0}$</td>
<td>$s_{3,1}$</td>
<td>$s_{3,2}$</td>
<td>$s_{3,3}$</td>
</tr>
</tbody>
</table>

First round only - generally it's $w_{i,r+c}$ where $c$ is the column and $r$ is the round.
Secret Key Systems - AES

**Key Schedule:** example for $N_k = 4$

<table>
<thead>
<tr>
<th>$w_{0,0}$, $w_{1,0}$, $w_{2,0}$, $w_{3,0}$</th>
<th>...</th>
<th>$w_{0,3}$, $w_{1,3}$, $w_{2,3}$, $w_{3,3}$</th>
<th>$w_{0,4}$, $w_{1,4}$, $w_{2,4}$, $w_{3,4}$</th>
</tr>
</thead>
</table>

1st round

2nd round

All rounds: $32*N_k$ bits for a round key
Secret Key Systems - AES

Key Expansion: example for $N_k = 4$

**RotWord ( )**: changes $[ a_0, a_1, a_2, a_3 ]$ to $[ a_3, a_2, a_1, a_0 ]$

**Rcon[i]**: the word $[ x^{i-1}, 0, 0, 0 ] \mod x^4 + 1$, where $x = 2$

**SubWord ( )**: maps each byte of $[ a_0, a_1, a_2, a_3 ]$ using S-Box values
**Secret Key Systems - AES**

**Rcon[i]:** also precomputed as a lookup table

<table>
<thead>
<tr>
<th>X</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>01 02 04 08 10 20 40 80 1b 36 6c d8 ab 4d 9a</td>
<td>2f 5e bc 63 c6 97 35 6a d4 b3 7d fa ef c5 91 39</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>72 e4 d3 bd 61 c2 9f 25 4a 94 33 66 cc 83 1d 3a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>74 e8 cb 8d 01 02 04 08 10 20 40 80 1b 36 6c d8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ab 4d 9a 2f 5e bc 63 c6 97 35 6a d4 b3 7d fa ef</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>c5 91 39 72 e4 d3 bd 61 c2 9f 25 4a 94 33 66 cc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>83 1d 3a 74 e8 cb 8d 01 02 04 08 10 20 40 80 1b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>36 6c d8 ab 4d 9a 2f 5e bc 63 c6 97 35 6a d4 b3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7d fa ef c5 91 39 72 e4 d3 bd 61 c2 9f 25 4a 94</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>33 66 cc 83 1d 3a 74 e8 cb 8d 01 02 04 08 10 20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>40 80 1b 36 6c d8 ab 4d 9a 2f 5e bc 63 c6 97 35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>6a d4 b3 7d fa ef c5 91 39 72 e4 d3 bd 61 c2 9f</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>25 4a 94 33 66 cc 83 1d 3a 74 e8 cb 8d 01 02 04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>08 10 20 40 80 1b 36 6c d8 ab 4d 9a 2f 5e bc 63</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>c6 97 35 6a d4 b3 7d fa ef c5 91 39 72 e4 d3 bd</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>61 c2 9f 25 4a 94 33 66 cc 83 1d 3a 74 e8 cb</td>
<td></td>
<td></td>
<td></td>
<td></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>
Secret Key Systems - AES

Key Expansion: example for \( N_k = 4 \)

**RotWord ( )**: changes \([ a_0, a_1, a_2, a_3 ]\) to \([ a_3, a_2, a_1, a_0 ]\)

**Rcon[i]**: the word \([ x^{i-1}, 0, 0, 0 ]\) \mod x^4 + 1, where \( x = 2 \)

**SubWord ( )**: maps each byte of \([ a_0, a_1, a_2, a_3 ]\) using S-Box values

**First Round**: the original key (16 bytes if \( N_k = 4 \))

**Other Rounds**: \([ w_{0,l}, w_{1,l}, w_{2,l}, w_{3,l} ]\)

\([ w_{3,l}, w_{2,l}, w_{1,l}, w_{0,l} ]\)  apply RotWord

\([ S(w_{0,l}), S(w_{1,l}), S(w_{2,l}), S(w_{3,l}) ]\)  apply SubWord

\([ S(w_{0,l}) \oplus \text{Rcon}[(l+1)/N_k], S(w_{1,l}), S(w_{2,l}), S(w_{3,l}) ]\)  use Rcon

\([ S(w_{0,l}) \oplus \text{Rcon}[(l+1)/N_k] \oplus w_{0,l}, S(w_{1,l}) \oplus w_{1,l}, S(w_{2,l}) \oplus w_{2,l}, S(w_{3,l}) \oplus w_{3,l} ]\]

\([ w_{0,l+1}, w_{1,l+1}, w_{2,l+1}, w_{3,l+1} ]\)  next key word
<table>
<thead>
<tr>
<th>i (dec)</th>
<th>temp</th>
<th>After RotWord()</th>
<th>After SubWord()</th>
<th>Rcon[i/Nk]</th>
<th>After XOR with Rcon</th>
<th>w[i-Nk]</th>
<th>w[i]=temp XOR w[i-Nk]</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>09cf4f3c</td>
<td>cf4f3c09</td>
<td>8a84eb01</td>
<td>01000000</td>
<td>8b84eb01</td>
<td>2b7e1516</td>
<td>a0fafe17</td>
</tr>
<tr>
<td>5</td>
<td>a0fafe17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>28aed2a6</td>
<td>88542cb1</td>
</tr>
<tr>
<td>6</td>
<td>88542cb1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>abf71588</td>
<td>23a33939</td>
</tr>
<tr>
<td>7</td>
<td>23a33939</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>09cf4f3c</td>
<td>2a6c7605</td>
</tr>
<tr>
<td>8</td>
<td>2a6c7605</td>
<td>6c76052a</td>
<td>50386be5</td>
<td>02000000</td>
<td>52386be5</td>
<td>a0fafe17</td>
<td>f2c295f2</td>
</tr>
<tr>
<td>9</td>
<td>f2c295f2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>88542cb1</td>
<td>7a96b943</td>
</tr>
<tr>
<td>10</td>
<td>7a96b943</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>23a33939</td>
<td>5935807a</td>
</tr>
<tr>
<td>11</td>
<td>5935807a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2a6c7605</td>
<td>7359f67f</td>
</tr>
<tr>
<td>12</td>
<td>7359f67f</td>
<td>59f67f73</td>
<td>cb42d28f</td>
<td>04000000</td>
<td>cf42d28f</td>
<td>f2c295f2</td>
<td>3d80477d</td>
</tr>
<tr>
<td>13</td>
<td>3d80477d</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7a96b943</td>
<td>4716fe3e</td>
</tr>
<tr>
<td>14</td>
<td>4716fe3e</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5935807a</td>
<td>1e237e44</td>
</tr>
<tr>
<td>15</td>
<td>1e237e44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7359f67f</td>
<td>6d7a883b</td>
</tr>
<tr>
<td>16</td>
<td>6d7a883b</td>
<td>7a883b6d</td>
<td>dac4e23c</td>
<td>08000000</td>
<td>d2c4e23c</td>
<td>3d80477d</td>
<td>ef44a541</td>
</tr>
<tr>
<td>17</td>
<td>ef44a541</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4716fe3e</td>
<td>a8525b7f</td>
</tr>
<tr>
<td>18</td>
<td>a8525b7f</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1e237e44</td>
<td>b671253b</td>
</tr>
<tr>
<td>19</td>
<td>b671253b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6d7a883b</td>
<td>db0bad00</td>
</tr>
<tr>
<td>20</td>
<td>db0bad00</td>
<td>0bad00db</td>
<td>2b9563b9</td>
<td>10000000</td>
<td>3b9563b9</td>
<td>ef44a541</td>
<td>d4d1c6f8</td>
</tr>
<tr>
<td>21</td>
<td>d4d1c6f8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>a8525b7f</td>
<td>7c839d87</td>
</tr>
<tr>
<td>22</td>
<td>7c839d87</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>b671253b</td>
<td>caf2b8bc</td>
</tr>
<tr>
<td>23</td>
<td>caf2b8bc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>db0bad00</td>
<td>11f915bc</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>11f915bc</td>
<td>f915bc11</td>
<td>99596582</td>
<td>20000000</td>
<td>b9596582</td>
<td>d4d1c6f8</td>
<td>6d88a37a</td>
</tr>
<tr>
<td>25</td>
<td>6d88a37a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>110b3efd</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>dbf98641</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>ca0093fd</td>
<td>0093fdca</td>
<td>63dc5474</td>
<td>40000000</td>
<td>23dc5474</td>
<td>6d88a37a</td>
<td>4e54f70e</td>
</tr>
<tr>
<td>29</td>
<td>4e54f70e</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>5f5fc9f3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>84a64fb2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>4ea6dc4f</td>
<td>a6dc4f4e</td>
<td>2486842f</td>
<td>80000000</td>
<td>a486842f</td>
<td>4e54f70e</td>
<td>ead27321</td>
</tr>
<tr>
<td>33</td>
<td>ead27321</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>b58dbad2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>312bf560</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>7f8d292f</td>
<td>8d292f7f</td>
<td>5da515d2</td>
<td>1b000000</td>
<td>46a515d2</td>
<td>ead27321</td>
<td>ac7766f3</td>
</tr>
<tr>
<td>37</td>
<td>ac7766f3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>19fadc21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>28d12941</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>575c006e</td>
<td>5c006e57</td>
<td>4a639f5b</td>
<td>36000000</td>
<td>7c639f5b</td>
<td>ac7766f3</td>
<td>d014f9a8</td>
</tr>
<tr>
<td>41</td>
<td>d014f9a8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>c9ee2589</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>e13f0cc8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Key: 2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C
# Secret Key Systems - AES

**Example:**

<table>
<thead>
<tr>
<th>Input</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>88</td>
<td>31</td>
<td>E0</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>5A</td>
<td>31</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>F6</td>
<td>30</td>
<td>98</td>
<td>07</td>
<td></td>
</tr>
<tr>
<td>A8</td>
<td>8D</td>
<td>A2</td>
<td>34</td>
<td></td>
</tr>
</tbody>
</table>

**Key:**

<table>
<thead>
<tr>
<th>Key</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2B</td>
<td>28</td>
<td>AB</td>
<td>09</td>
<td></td>
</tr>
<tr>
<td>7E</td>
<td>AE</td>
<td>F7</td>
<td>CF</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>D2</td>
<td>15</td>
<td>4F</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>A6</td>
<td>88</td>
<td>3C</td>
<td></td>
</tr>
</tbody>
</table>
Secret Key Systems - AES

Example:

Input:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>88</td>
<td>31</td>
<td>E0</td>
</tr>
<tr>
<td>43</td>
<td>5A</td>
<td>31</td>
<td>37</td>
</tr>
<tr>
<td>F6</td>
<td>30</td>
<td>98</td>
<td>07</td>
</tr>
<tr>
<td>A8</td>
<td>8D</td>
<td>A2</td>
<td>34</td>
</tr>
</tbody>
</table>

Key:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2B</td>
<td>28</td>
<td>AB</td>
<td>09</td>
</tr>
<tr>
<td>7E</td>
<td>AE</td>
<td>F7</td>
<td>CF</td>
</tr>
<tr>
<td>15</td>
<td>D2</td>
<td>15</td>
<td>4F</td>
</tr>
<tr>
<td>16</td>
<td>A6</td>
<td>88</td>
<td>3C</td>
</tr>
</tbody>
</table>

Key:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>A0</td>
<td>9A</td>
<td>E9</td>
</tr>
<tr>
<td>3D</td>
<td>F4</td>
<td>C6</td>
<td>F8</td>
</tr>
<tr>
<td>E3</td>
<td>E2</td>
<td>8D</td>
<td>48</td>
</tr>
<tr>
<td>BE</td>
<td>2B</td>
<td>2A</td>
<td>08</td>
</tr>
</tbody>
</table>

beginning of first round - input placed into the state and key has been added to state
Secret Key Systems - AES

Example:

<table>
<thead>
<tr>
<th>State:</th>
<th>19</th>
<th>A0</th>
<th>9A</th>
<th>E9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3D</td>
<td>F4</td>
<td>C6</td>
<td>F8</td>
</tr>
<tr>
<td></td>
<td>E3</td>
<td>E2</td>
<td>8D</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>BE</td>
<td>2B</td>
<td>2A</td>
<td>08</td>
</tr>
</tbody>
</table>

S-Box

<table>
<thead>
<tr>
<th>S-Box</th>
<th>D4</th>
<th>E0</th>
<th>B8</th>
<th>1E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>27</td>
<td>BF</td>
<td>B4</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>98</td>
<td>5D</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>AE</td>
<td>F1</td>
<td>E5</td>
<td>30</td>
</tr>
</tbody>
</table>

ShiftRows

<table>
<thead>
<tr>
<th>ShiftRows</th>
<th>D4</th>
<th>E0</th>
<th>B8</th>
<th>1E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>04</td>
<td>E0</td>
<td>48</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>66</td>
<td>CB</td>
<td>F8</td>
<td>06</td>
</tr>
<tr>
<td></td>
<td>81</td>
<td>19</td>
<td>D3</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>E5</td>
<td>9A</td>
<td>7A</td>
<td>4C</td>
</tr>
</tbody>
</table>

MixColumns
Secret Key Systems - AES

Performance Notes:

1. Many operations are table look ups so they are fast
2. Parallelism can be exploited
3. Key expansion only needs to be done one time until the key is changed
4. The S-box minimizes the correlation between input and output bits
5. There are no known weak keys
Secret Key Systems - AES

Attacks:

- **Extended Sparse Linearization** -
  Derive a system of quadratic simultaneous equations and solve,
  128 bit key: 8000 quadratic equations, 1600 variables
  256 bit key: 22400 quadratic equations, 4480 variables
  given plaintext, to get the key – not practical although $2^{100}$ vs. $2^{128}$
Secret Key Systems - AES

Attacks:

- **Extended Sparse Linearization** -
  Derive a system of quadratic simultaneous equations and solve, 
  128 bit key: 8000 quadratic equations, 1600 variables 
  256 bit key: 22400 quadratic equations, 4480 variables 
  given plaintext, to get the key – not practical although $2^{100}$ vs. $2^{128}$

- **Related Key** -
  Attacker may be able to observe behavior of cipher in the case of 
  several keys, not initially known, but with some understanding 
  of the mathematical relationship connecting the keys - e.g. 
  the number of 1s equals the number of 0s – $2^{70}$ time for an 11 
  round version of 256-bit AES but attack on full version is not reported.
Secret Key Systems - AES

Attacks:

- **Extended Sparse Linearization** -
  Derive a system of quadratic simultaneous equations and solve,
  128 bit key: 8000 quadratic equations, 1600 variables
  256 bit key: 22400 quadratic equations, 4480 variables
  given plaintext, to get the key – not practical although $2^{100}$ vs. $2^{128}$

- **Related Key** -
  Attacker may be able to observe behavior of cipher in the case of
  several keys, not initially known, but with some understanding
  of the mathematical relationship connecting the keys - e.g.
  the number of 1s equals the number of 0s – $2^{70}$ time for an 11
  round version of 256-bit AES but attack on full version is not
  reported.

- **Titled Attack** -
  If attacker can stop the execution of encryption, apply a “difference”
  to the state, and roll back the encryption. $2^{32}$ vs. $2^{128}$
Secret Key Systems - AES

Attacks:

- **Extended Sparse Linearization** -
  Derive a system of quadratic simultaneous equations and solve,
  128 bit key: 8000 quadratic equations, 1600 variables
  256 bit key: 22400 quadratic equations, 4480 variables
  given plaintext, to get the key – not practical although $2^{100}$ vs. $2^{128}$

- **Related Key** -
  Attacker may be able to observe behavior of cipher in the case of
  several keys, not initially known, but with some understanding
  of the mathematical relationship connecting the keys - e.g.
  the number of 1s equals the number of 0s – $2^{70}$ time for an 11
  round version of 256-bit AES but attack on full version is not
  reported.

- **Titled Attack** -
  If attacker can stop the execution of encryption, apply a “difference”
  to the state, and roll back the encryption. $2^{32}$ vs. $2^{128}$

- **Side Channel** -
  Vulnerable in some implementations – later in the course
Secret Key Systems - AES

Number of rounds:

<table>
<thead>
<tr>
<th>$N_k$</th>
<th>$N_b$</th>
<th>4</th>
<th>6</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>12</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td></td>
</tr>
</tbody>
</table>