Advanced Encryption Standard (AES)

Cryptography – CS 507
Erkay Savas
Sabanci University
erkays@sabanciuniv.edu
History of Rijndael

- Successor to DES
- The AES selection is administered by NIST
- Unlike DES, AES selection was an open process.
  - 1997, NIST called for candidates to replace DES.
  - Requirements were
    - Block cipher with 128-bit block size
    - Support for 128, 192, 256 bits of key sizes
    - Efficient software and hardware implementation.
  - Cryptographic community was asked to comment on five finalists, MARS(IBM), RC6(RSA), Rijndael, Serpent, Twofish.
  - NIST chose Rijndael as AES in 2000.
- Likely to be the most commonly used algorithm in the next decade.
- See http://www.nist.gov/aes for more information
# Speeds of the five finalists

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Pentium Pro 200 Mhz Mbit/s</th>
<th>FPGA hardware Gbit/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>MARS</td>
<td>69</td>
<td>-</td>
</tr>
<tr>
<td>RC6</td>
<td>105</td>
<td>2.4</td>
</tr>
<tr>
<td>Rijndael</td>
<td>71</td>
<td>1.9</td>
</tr>
<tr>
<td>Serpent</td>
<td>27</td>
<td>4.9</td>
</tr>
<tr>
<td>Twofish</td>
<td>95</td>
<td>1.6</td>
</tr>
</tbody>
</table>
Rijndael Overview

- Block size is also variable (128/192/256)
- # of rounds is a function of key length:

<table>
<thead>
<tr>
<th>Key length (in bits)</th>
<th>#of rounds $n_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>10</td>
</tr>
<tr>
<td>192</td>
<td>12</td>
</tr>
<tr>
<td>256</td>
<td>14</td>
</tr>
</tbody>
</table>
Rijndael overview

• Rijndael is not a Feistel cipher.
  – Feistel ciphers do not encrypt the whole block in each iteration. This explains why Rijndael has fewer # of rounds.

• Rijndael has three basic steps (or so called layers):
  – **Key Addition Layer**: XORing the block with the round key.
  – **Byte Substitution Layer**: 8-by-8 substitution (s-box). Nonlinear operation (confusion).
  – **Diffusion Layer**: provides the diffusion of the bits of a block. Linear diffusion layer.
    • ShiftRow Layer
    • MixColumn Layer
Rijndael Encryption Block

11/7/2002 Erkay Savas

The Layers

- Throughout the discussions here we will assume the block and key lengths are fixed to 128-bit (16 bytes).
- 16 bytes (128 bit) are arranged into as 4×4 matrix

\[
\begin{pmatrix}
a_{0,0}, a_{1,0}, a_{2,0}, a_{3,0}, a_{0,1}, a_{1,1}, a_{2,1}, a_{3,1} \\
a_{0,2}, a_{1,2}, a_{2,2}, a_{3,2} \\
a_{0,3}, a_{1,3}, a_{2,3}, a_{3,3}
\end{pmatrix}
\]

- Each matrix entry can be thought an element of GF(2^8) with \(x^8 + x^4 + x^3 + x + 1\).
  - We will occasionally do arithmetic in GF(2^8).
  - Addition, multiplication, inversion.
The Byte Substitution Layer

- Each byte in the matrix is changed to another byte by the following operations:
  - Each byte $A$ is considered an element of $\text{GF}(2^8)$, $A(x)$.
  - Find the multiplicative inverse of $A(x)$, $T(x)=A^{-1}(x)$.
  - Apply the affine transformation defined by

\[
\begin{pmatrix}
    b_0 \\
    b_1 \\
    b_2 \\
    b_3 \\
    b_4 \\
    b_5 \\
    b_6 \\
    b_7 \\
\end{pmatrix} =
\begin{pmatrix}
    1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
    0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
    0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
    0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
    1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
    1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
    1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
    1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
\end{pmatrix}
\begin{pmatrix}
    t_0 \\
    t_1 \\
    t_2 \\
    t_3 \\
    t_4 \\
    t_5 \\
    t_6 \\
    t_7 \\
\end{pmatrix} +
\begin{pmatrix}
    0 \\
    1 \\
    1 \\
    0 \\
    0 \\
    0 \\
    1 \\
    1 \\
\end{pmatrix}
The Byte Substitution Layer

- The result is another 4×4 matrix whose entries are bytes.

\[
\begin{pmatrix}
  b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\
  b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\
  b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\
  b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3}
\end{pmatrix}
\]

- You can use a 8-by-8 table whose entries are bytes in order to implement this layer.
The Shift Row Layer

- Four rows of the matrix are shifted cyclically to the left by offsets of 0, 1, 2, 3.

\[
\begin{pmatrix}
c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\
c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} \\
c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\
c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3}
\end{pmatrix}
= 
\begin{pmatrix}
b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\
b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\
b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\
b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3}
\end{pmatrix}
\]
The Mix Column Layer

$$\begin{bmatrix}
d_{0,0} & d_{0,1} & d_{0,2} & d_{0,3} \\
d_{1,0} & d_{1,1} & d_{1,2} & d_{1,3} \\
d_{2,0} & d_{2,1} & d_{2,2} & d_{2,3} \\
d_{3,0} & d_{3,1} & d_{3,2} & d_{3,3}
\end{bmatrix} =
\begin{bmatrix}
02 & 03 & 01 & 01 \\
01 & 02 & 03 & 01 \\
01 & 01 & 02 & 03 \\
03 & 01 & 01 & 02
\end{bmatrix} \cdot
\begin{bmatrix}
c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\
c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} \\
c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\
c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3}
\end{bmatrix}$$

- $02 = 0000 \ 0010 = x$
- $03 = 0000 \ 0011 = x+1$
- The Shift Row and the Mix Column Layers performs linear transformations, i.e.
DIFF($A$) $\oplus$ DIFF($B$) = DIFF($A$ $\oplus$ $B$)
The Round Key Addition

- A simple XORing operation

\[
\begin{pmatrix}
  e_{0,0} & e_{0,1} & e_{0,2} & e_{0,3} \\
  e_{1,0} & e_{1,1} & e_{1,2} & e_{1,3} \\
  e_{2,0} & e_{2,1} & e_{2,2} & e_{2,3} \\
  e_{3,0} & e_{3,1} & e_{3,2} & e_{3,3}
\end{pmatrix} = \begin{pmatrix}
  d_{0,0} & d_{0,1} & d_{0,2} & d_{0,3} \\
  d_{1,0} & d_{1,1} & d_{1,2} & d_{1,3} \\
  d_{2,0} & d_{2,1} & d_{2,2} & d_{2,3} \\
  d_{3,0} & d_{3,1} & d_{3,2} & d_{3,3}
\end{pmatrix} \oplus \begin{pmatrix}
  k_{0,0} & k_{0,1} & k_{0,2} & k_{0,3} \\
  k_{1,0} & k_{1,1} & k_{1,2} & k_{1,3} \\
  k_{2,0} & k_{2,1} & k_{2,2} & k_{2,3} \\
  k_{3,0} & k_{3,1} & k_{3,2} & k_{3,3}
\end{pmatrix}
\]

- Matrix whose entries are \( e_{i,j} \) is the output of the round
The Key schedule

- The original key consists of 128 bits, which are arranged into a $4 \times 4$ matrix of bytes.
- This matrix is expanded by adjoining 40 more columns, as follows:
  - Label the first four (original columns) as $W(0)$, $W(1)$, $W(2)$, $W(3)$.
  - The new columns are generated recursively:
    - If $i$ is a multiple of 4 then $W(i) = W(i-4) \oplus W(i-1)$
    - Otherwise, $W(i) = W(i-4) \oplus T(W(i-1))$
    - Where $T(W(i-1))$ is obtained as follows
The Key schedule: $T(W(i-1))$

- Let the elements of $W(i-1)$ be $a,b,c,d$.
- Shift these cyclically to obtain $b,c,d,a$.
- Now transform each byte using *Byte substitution* into $e,f,g,h$.
- Finally, compute the round constant $r(i) = x^{(i-4)/4}$ in $\text{GF}(2^8)$.
- Then, $T(W(i-1))$ is the column vector $(e \oplus r(i), f, g, h)$.
- The **round key** for the $i$th round consists of the columns $W(4i)$, $W(4i+1)$, $W(4i+2)$, $W(4i+3)$. 
Decryption

- Rijndael is not Feistel cipher; thus each layer must actually be inverted.
- Operations in each layer are invertible:
  - \text{InvByteSub}
  - \text{InvShiftRow}
    - Shift right instead of left.
  - \text{InvMixColumn}
    - The inverse of MixColumn exists because 4×4 matrix used in MixColumn is invertible. InvMixColumn matrix

\[
\begin{pmatrix}
0E & 0B & 0D & 09 \\
09 & 0E & 0B & 0D \\
0D & 09 & 0E & 0B \\
0B & 0D & 09 & 0E \\
\end{pmatrix}
\]

\[OE = 00001110 \Rightarrow x^3 + x^2 + x\]
Rijndael Decryption

Inverse of rounds 1 \ldots n_r - 1

Round $n_r$

Key Addition Layer
InvShiftRow Layer
InvByteSub Layer

Key Addition Layer
InvMixColumn Layer
InvShiftRow Layer
InvByteSub Layer

$y$

$x$
Final Remarks

- In every round, each bit in the block are treated uniformly
  - This has the effect of diffusing the input bits faster
  - After two rounds each of the 128 output bits depends on each of the 128 input bits.
- S-box is constructed using a very simple algebraic mapping, $x \rightarrow x^{-1}$ in GF($2^8$).
  - The mapping is highly nonlinear.
  - Its simplicity removes any suspicions about a certain trapdoor which is believed to exist in DES for years.
- The MixColumn layer causes diffusion in the byte level.
- Key scheduling also utilizes highly nonlinear Byte Substitution mapping.
- No known attacks are better than brute force for seven or more rounds (Rijndael makes use of at least 10 rounds).