

# AES

- Overview of the AES algorithm
- Internal structure of AES
  - Byte Substitution layer
  - Diffusion layer
  - Key Addition layer
  - Key schedule
- Decryption
- Practical issues

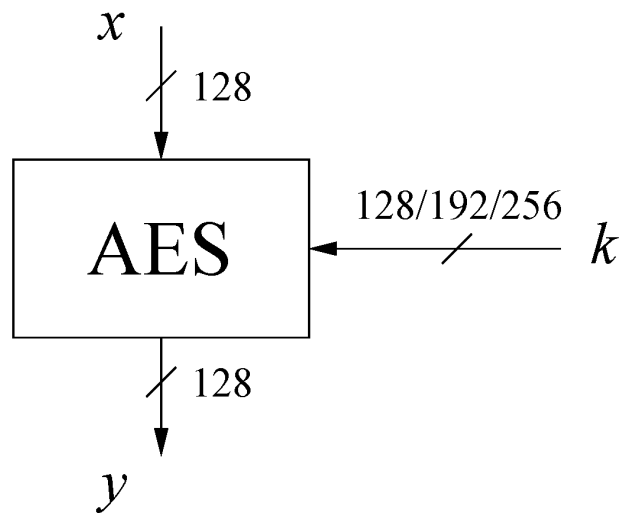
## ■ Some Basic Facts

- AES is the most widely used symmetric cipher today
- The algorithm for AES was chosen by the US *National Institute of Standards and Technology* (NIST) in a multi-year selection process
- The requirements for all AES candidate submissions were:
  - Block cipher with **128-bit block size**
  - **Three supported key lengths**: 128, 192 and 256 bit
  - Security relative to other submitted algorithms
  - **Efficiency** in software and hardware

## ■ Chronology of the AES Selection

- The need for a new block cipher announced by NIST in January, 1997
- 15 candidates algorithms accepted in August, 1998
- 5 finalists announced in August, 1999:
  - *Mars* – IBM Corporation
  - *RC6* – RSA Laboratories
  - *Rijndael* – J. Daemen & V. Rijmen
  - *Serpent* – Eli Biham et al.
  - *Twofish* – B. Schneier et al.
- In October 2000, *Rijndael* was chosen as the AES
- AES was formally approved as a US federal standard in November 2001

## ■ AES: Overview

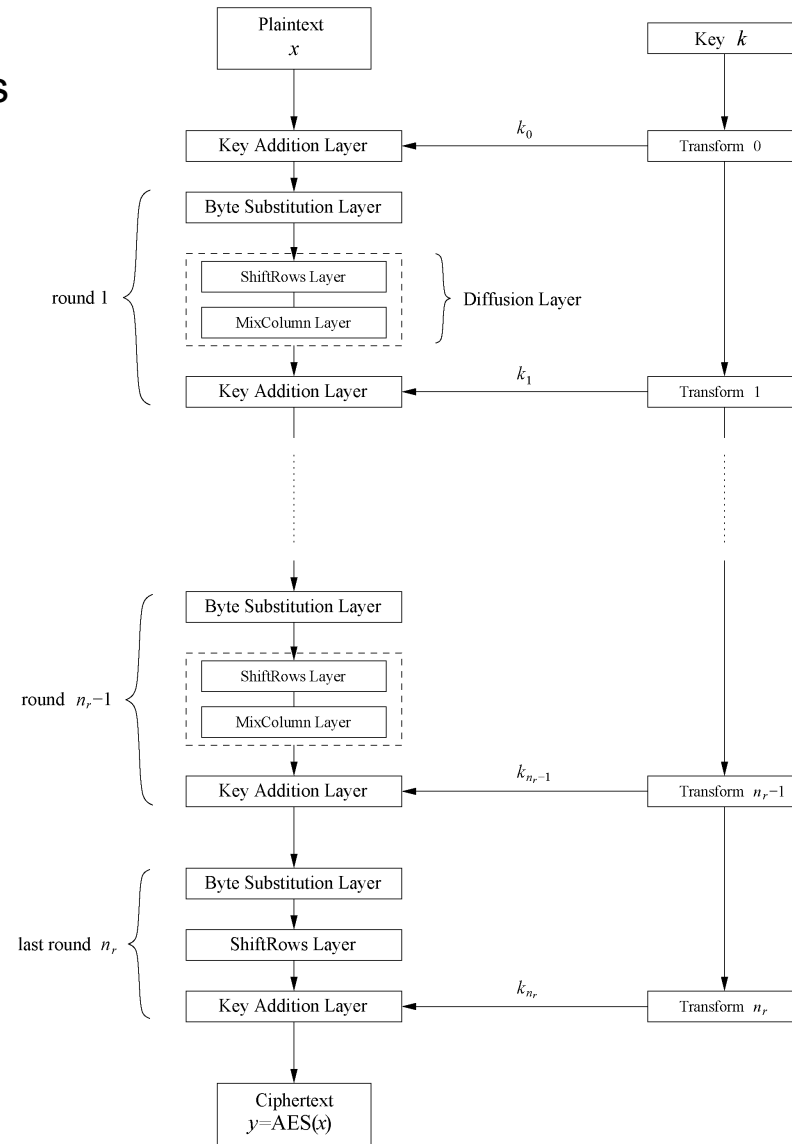


The number of rounds depends on the chosen key length:

Key length (bits)	Number of rounds
128	10
192	12
256	14

## ■ AES: Overview

- Iterated cipher with 10/12/14 rounds
- Each round consists of “Layers”



## ■ Internal Structure of AES

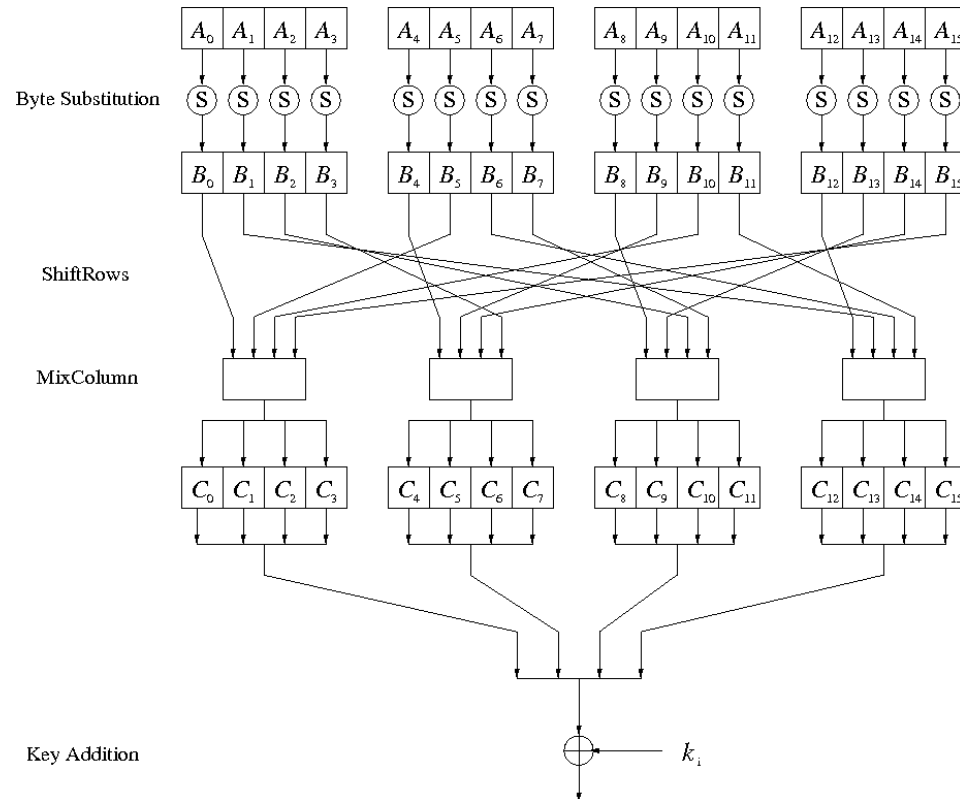
- AES is a byte-oriented cipher
- The state  $A$  (i.e., the 128-bit data path) can be arranged in a 4x4 matrix:

$A_0$	$A_4$	$A_8$	$A_{12}$
$A_1$	$A_5$	$A_9$	$A_{13}$
$A_2$	$A_6$	$A_{10}$	$A_{14}$
$A_3$	$A_7$	$A_{11}$	$A_{15}$

with  $A_0, \dots, A_{15}$  denoting the 16-byte input of AES

## ■ Internal Structure of AES

- Round function for rounds  $1, 2, \dots, n_{r-1}$ :




- Note: In the last round, the MixColumn transformation is omitted

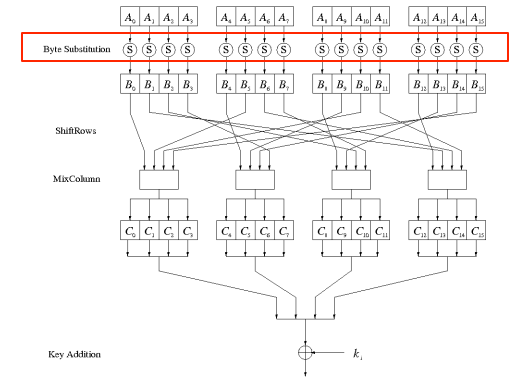
## ■ Byte Substitution Layer

- The Byte Substitution layer consists of 16 **S-Boxes** with the following properties:

The S-Boxes are

- identical**
- the only **nonlinear** elements of AES, i.e.,  
 $\text{ByteSub}(A_i) + \text{ByteSub}(A_j) \neq \text{ByteSub}(A_i + A_j)$ , for  $i, j = 0, \dots, 15$
- bijective**, i.e., there exists a one-to-one mapping of input and output bytes
-  S-Box can be uniquely reversed

- In software implementations, the S-Box is usually realized as a lookup table

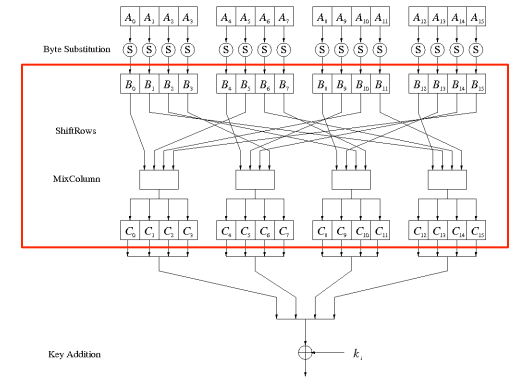




## ■ Diffusion Layer

The Diffusion layer

- provides diffusion over all input state bits
- consists of two sublayers:
  - **ShiftRows Sublayer**: Permutation of the data on a byte level
  - **MixColumn Sublayer**: Matrix operation which combines (“mixes”) blocks of four bytes
- performs a linear operation on state matrices  $A$ ,  $B$ , i.e.,
$$\text{DIFF}(A) + \text{DIFF}(B) = \text{DIFF}(A + B)$$



## ■ ShiftRows Sublayer

- Rows of the state matrix are shifted cyclically:

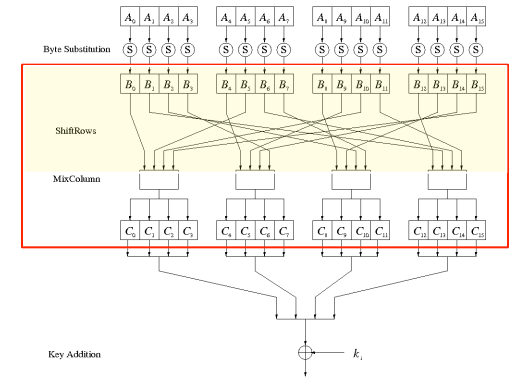
Input matrix

$B_0$	$B_4$	$B_8$	$B_{12}$
$B_1$	$B_5$	$B_9$	$B_{13}$
$B_2$	$B_6$	$B_{10}$	$B_{14}$
$B_3$	$B_7$	$B_{11}$	$B_{15}$

Output matrix

$B_0$	$B_4$	$B_8$	$B_{12}$
$B_5$	$B_9$	$B_{13}$	$B_1$
$B_{10}$	$B_{14}$	$B_2$	$B_6$
$B_{15}$	$B_3$	$B_7$	$B_{11}$

- no shift
- ← one position left shift
- ← two positions left shift
- ← three positions left shift



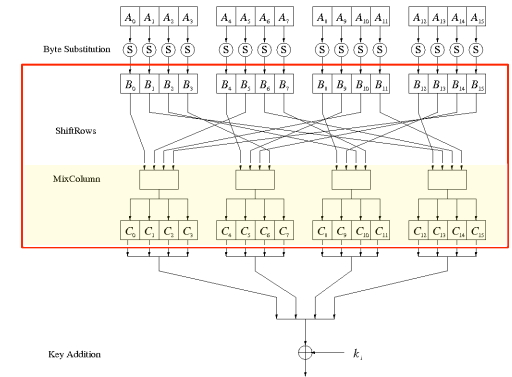
## ■ MixColumn Sublayer

- Linear transformation which mixes each column of the state matrix
- Each 4-byte column is considered as a vector and multiplied by a fixed 4x4 matrix, e.g.,

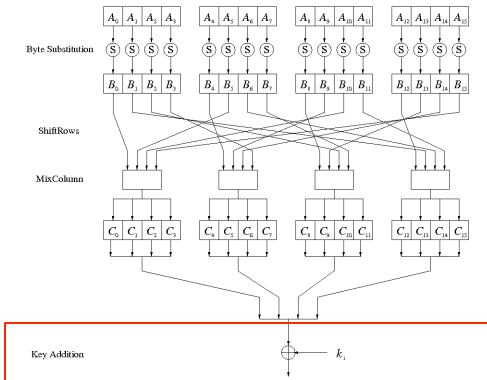
$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} B_0 \\ B_5 \\ B_{10} \\ B_{15} \end{pmatrix}$$

where 01, 02 and 03 are given in hexadecimal notation

- All arithmetic is done in the Galois field  $GF(2^8)$



## ■ Key Addition Layer



- Inputs:
  - 16-byte state matrix  $C$
  - 16-byte subkey  $k_i$
- Output:  $C \boxplus k_i$
- The subkeys are generated in the key schedule

## ■ Key Schedule

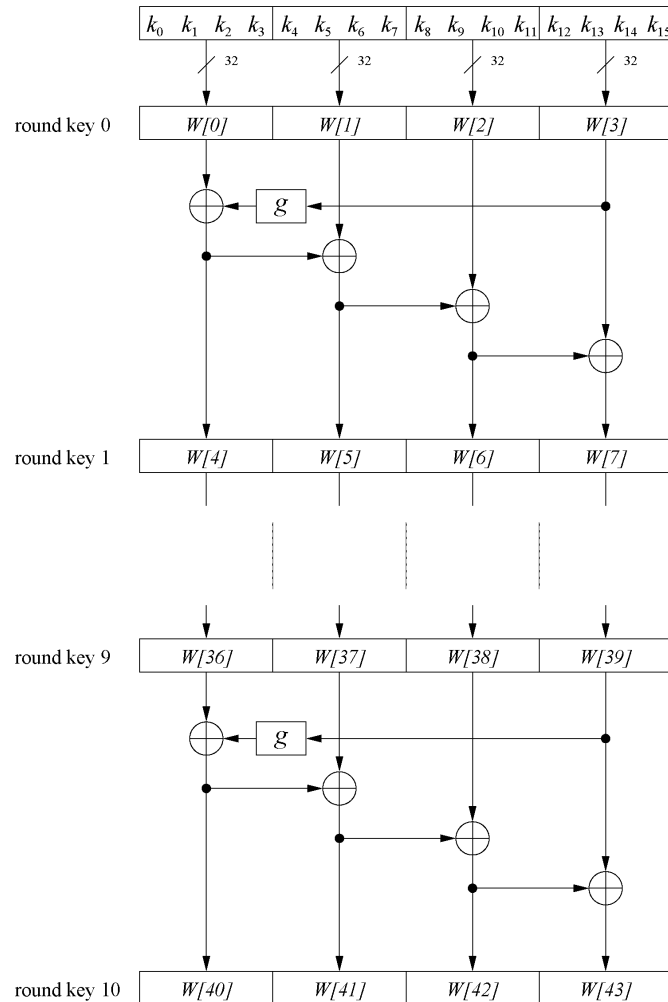
- Subkeys are derived recursively from the original 128/192/256-bit input key
- Each round has 1 subkey, plus 1 subkey at the beginning of AES

Key length (bits)	Number of subkeys
128	11
192	13
256	15

- Key whitening: Subkey is used both at the input and output of AES  
     $\# \text{ subkeys} = \# \text{ rounds} + 1$
- There are different key schedules for the different key sizes

## ■ Key Schedule

Example: Key schedule for 128-bit key AES



- Word-oriented: 1 word = 32 bits
- 11 subkeys are stored in  $W[0] \dots W[3]$ ,  $W[4] \dots W[7]$ , ...,  $W[40] \dots W[43]$
- First subkey  $W[0] \dots W[3]$  is the original AES key

## ■ Key Schedule

- Function  $g$  rotates its four input bytes and performs a bitwise S-Box substitution  
⊗ nonlinearity

- The round coefficient  $RC$  is only added to the leftmost byte and varies from round to round:

$$RC[1] = x^0 = (00000001)_2$$

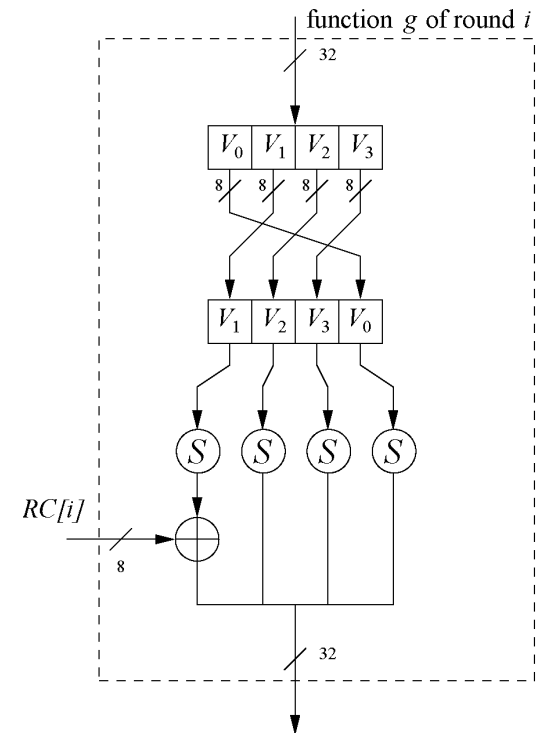
$$RC[2] = x^1 = (00000010)_2$$

$$RC[3] = x^2 = (00000100)_2$$

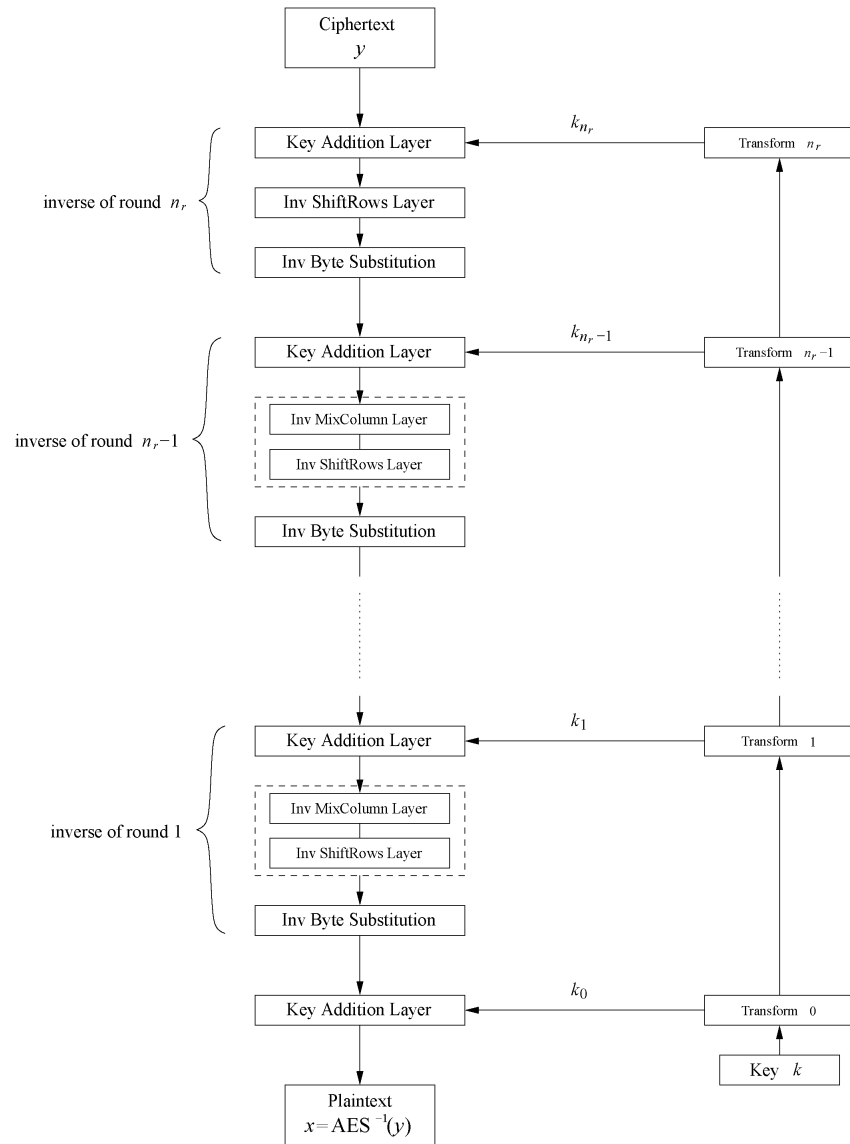
...

$$RC[10] = x^9 = (00110110)_2$$

- $x^i$  represents an element in a Galois field



## ■ Decryption



- AES is not based on a Feistel network

☒ All layers must be inverted for decryption:

- MixColumn layer → **Inv MixColumn layer**
- ShiftRows layer → **Inv ShiftRows layer**
- Byte Substitution layer → **Inv Byte Substitution layer**
- Key Addition layer is its own inverse



## ■ Decryption

- **Inv MixColumn layer:**

- To reverse the MixColumn operation, each column of the state matrix  $C$  must be multiplied with the **inverse of the 4x4 matrix**, e.g.,

$$\begin{pmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{pmatrix} = \begin{pmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{pmatrix} \cdot \begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix}$$

where 09, 0B, 0D and 0E are given in hexadecimal notation

- Again, all arithmetic is done in the Galois field  $GF(2^8)$

## ■ Decryption

- **Inv ShiftRows layer:**

- All rows of the state matrix  $B$  are shifted to the opposite direction:

Input matrix

$B_0$	$B_4$	$B_8$	$B_{12}$
$B_1$	$B_5$	$B_9$	$B_{13}$
$B_2$	$B_6$	$B_{10}$	$B_{14}$
$B_3$	$B_7$	$B_{11}$	$B_{15}$

Output matrix

$B_0$	$B_4$	$B_8$	$B_{12}$
$B_{13}$	$B_1$	$B_5$	$B_9$
$B_{10}$	$B_{14}$	$B_2$	$B_6$
$B_7$	$B_{11}$	$B_{15}$	$B_3$

no shift

→ one position right shift

→ two positions right shift

→ three positions right shift

## ■ Decryption

- **Inv Byte Substitution layer:**

- Since the S-Box is bijective, it is possible to construct an inverse, such that

$$A_i = S^{-1}(B_i) = S^{-1}(S(A_i))$$

☐ The inverse S-Box is used for decryption. It is usually realized as a lookup table

- **Decryption key schedule:**

- Subkeys are needed in reversed order (compared to encryption)
- In practice, for encryption and decryption, the same key schedule is used. This requires that all subkeys must be computed before the encryption of the first block can begin

## ■ Implementation in Software

- One requirement of AES was the possibility of an efficient software implementation
- Straightforward implementation is well suited for 8-bit processors (e.g., smart cards), but inefficient on 32-bit or 64-bit processors
- A more sophisticated approach: Merge all round functions (except the key addition) into one table look-up
  - This results in four tables with 256 entries, where each entry is 32 bits wide
  - One round can be computed with 16 table look-ups
- Typical SW speeds are more than 1.6 Gbit/s on modern 64-bit processors

## ■ Security

- **Brute-force attack:** Due to the key length of 128, 192 or 256 bits, a brute-force attack is not possible
- **Analytical attacks:** There is no analytical attack known that is better than brute-force
- **Side-channel attacks:**
  - Several side-channel attacks have been published
  - Note that side-channel attacks do not attack the underlying algorithm but the implementation of it

## ■ Lessons Learned

- AES is a modern block cipher which supports three key lengths of 128, 192 and 256 bit. It provides excellent long-term security against brute-force attacks.
- AES has been studied intensively since the late 1990s and no attacks have been found that are better than brute-force.
- AES is not based on Feistel networks. Its basic operations use Galois field arithmetic and provide strong diffusion and confusion.
- AES is part of numerous open standards such as IPsec or TLS, in addition to being the mandatory encryption algorithm for US government applications. It seems likely that the cipher will be the dominant encryption algorithm for many years to come.
- AES is efficient in software and hardware.