WILFRID LAURIER UNIVERSITY
LAURIER
Inspiring Lives.

CP 460 - Applied Cryptography

The Advanced Encryption Standard (AES)

Department of Physics and Computer Science
Faculty of Science, Waterloo

Abbas Yazdinejad, Ph.D.

Fall 2024

Content of this Chapter

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

Content of this Chapter

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

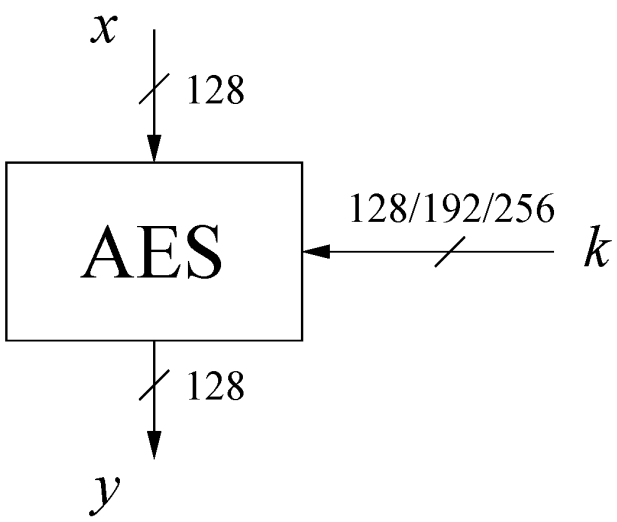
Some Basic Facts

- 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

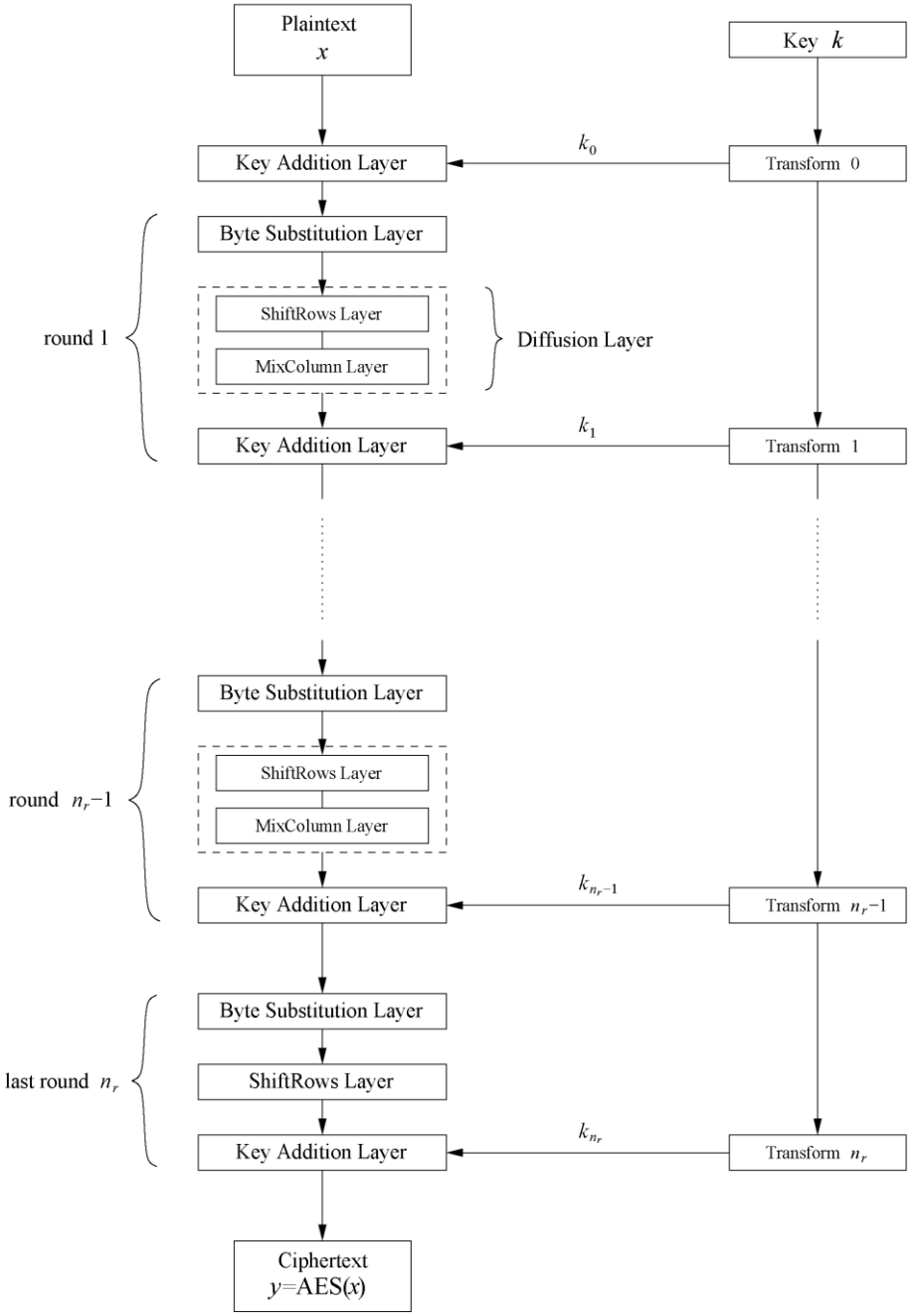


The number of rounds (applying transformations such as substitution, permutation, and mixing of data) depends on the chosen key length:

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”



Content of this Chapter

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

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:

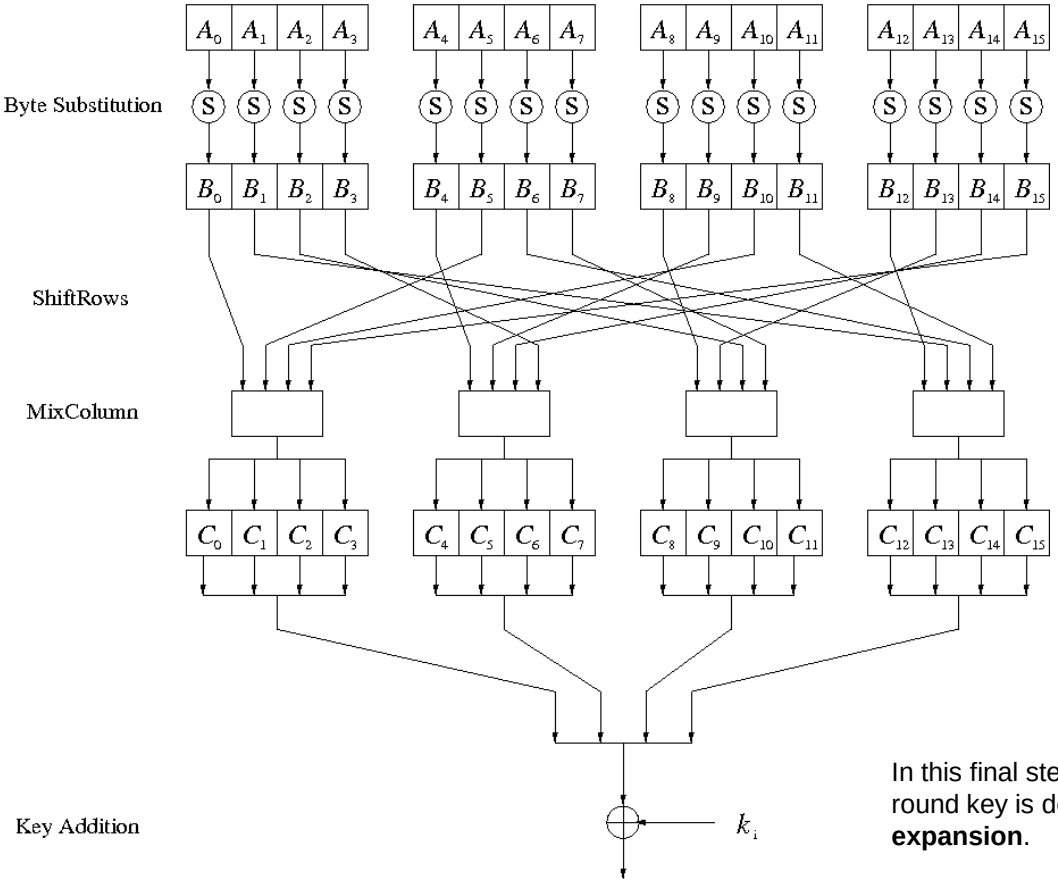
A_0	A_4	A_8	A_{12}
A_1	A_5	A_9	A_{13}
A_2	A_6	A_{10}	A_{14}
A_3	A_7	A_{11}	A_{15}

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

The **state matrix** is a key concept in AES, where the 128-bit input data is divided into 16 bytes and arranged in a 4-row by 4-column matrix. Each byte is represented as a two-digit hexadecimal number.

Internal Structure of AES

- Round function for rounds 1,2,…,n_{r-1}:



This is a **non-linear byte substitution** step. Each byte in the state matrix is replaced using a fixed 16x16 lookup table called the **S-box**.

In **ShiftRows**, the rows of the state matrix are shifted **cyclically** to the left by a certain number of positions.

In **MixColumns**, each column of the state matrix is transformed independently.

In this final step, the current **round key** is XORed with the state matrix. Each round key is derived from the original key through a process called **key expansion**.

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

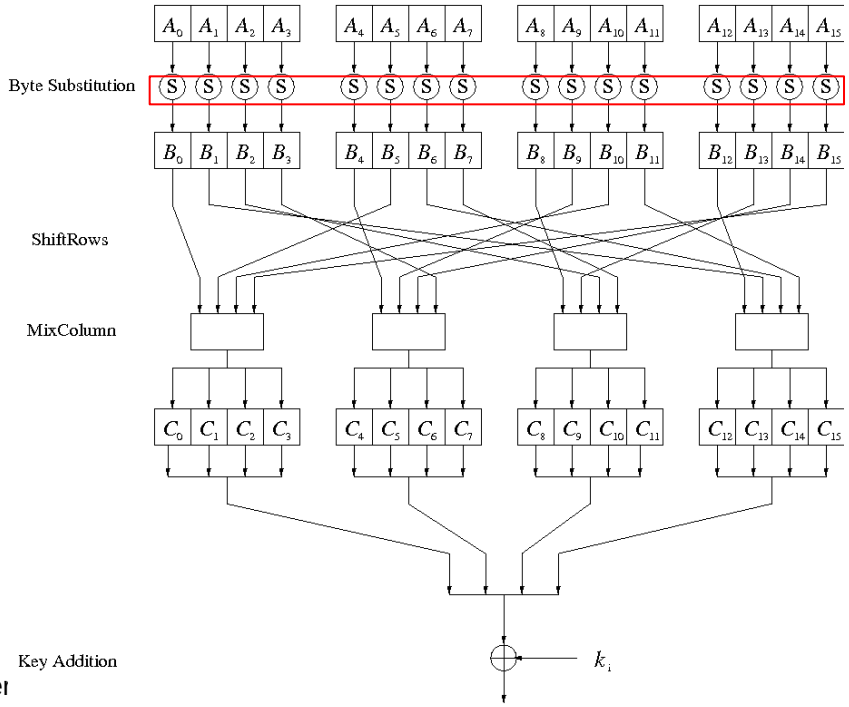
Byte Substitution Layer

- 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.,
    ByteSub(A_i) + ByteSub(A_j) ≠ ByteSub(A_i + A_j), for i,j = 0,…,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

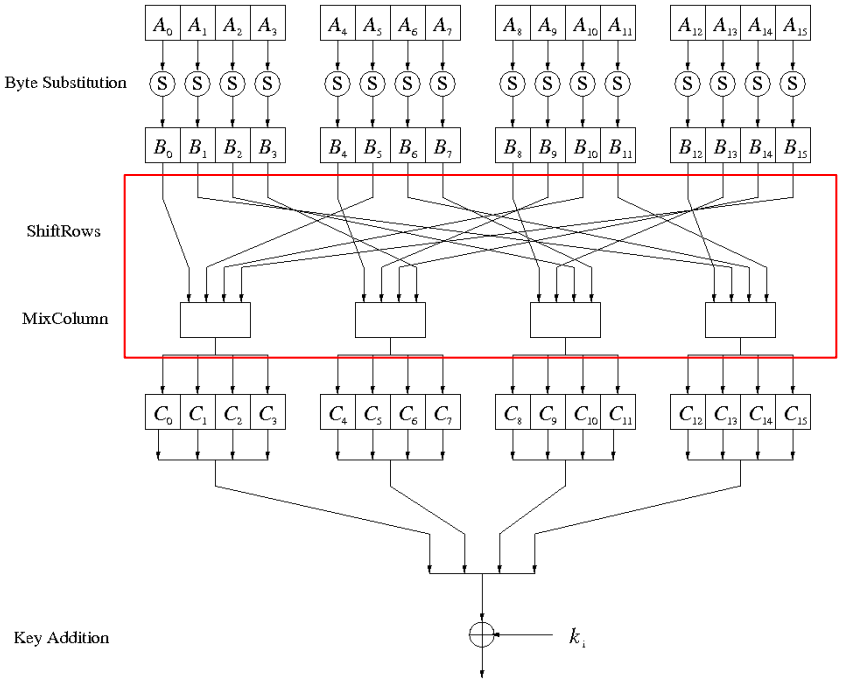
- consists of two sublayers:
  - **ShiftRows Sublayer**: Permutation of the data on a byte level
  - **MixColumn Sublayer**: Matrix operation which combines ("mixes") blocks of four bytes

- performs a linear operation on state matrices *A*, *B*, i.e.,

$$\text{DIFF}(A) + \text{DIFF}(B) = \text{DIFF}(A + B)$$

The diffusion layer (comprising **ShiftRows** and **MixColumns**) is **linear**.

Together, these two operations ensure that AES provides excellent diffusion, spreading the effects of each input bit across the entire output, making the cipher more resistant to cryptanalytic attacks like differential cryptanalysis.



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

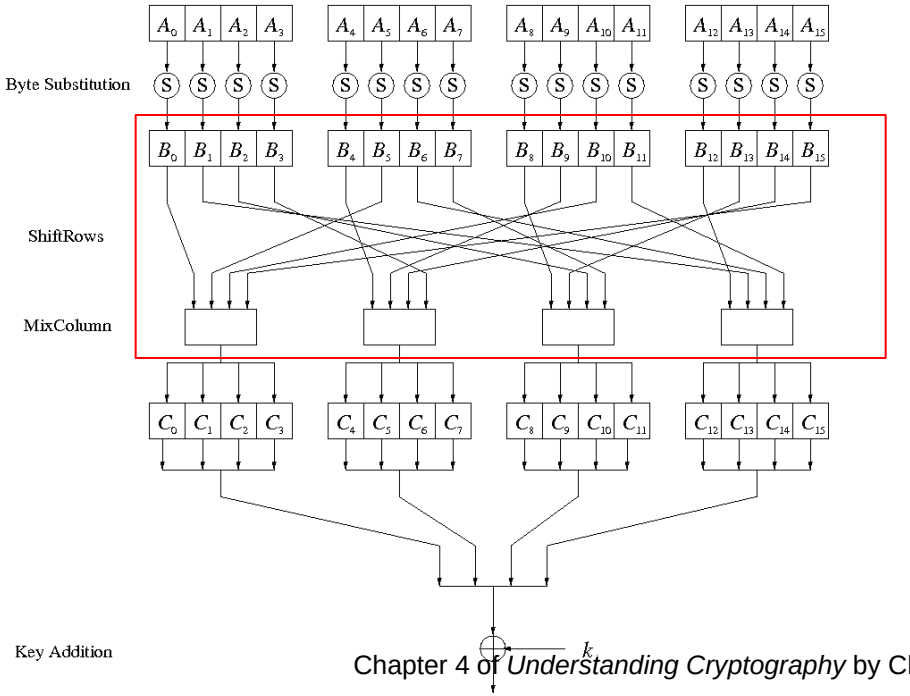
B_0	B_4	B_8	B_{12}
B_5	B_9	B_{13}	B_1
B_{10}	B_{14}	B_2	B_6
B_{15}	B_3	B_7	B_{11}

no shift

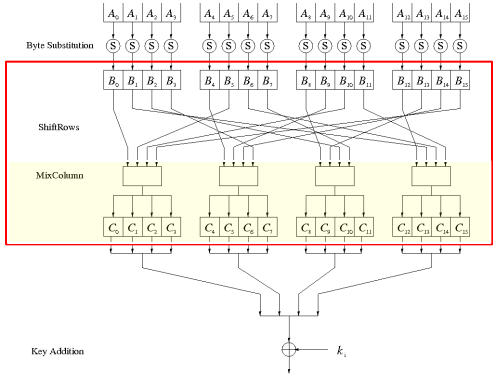
← one position left shift

← 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.,

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

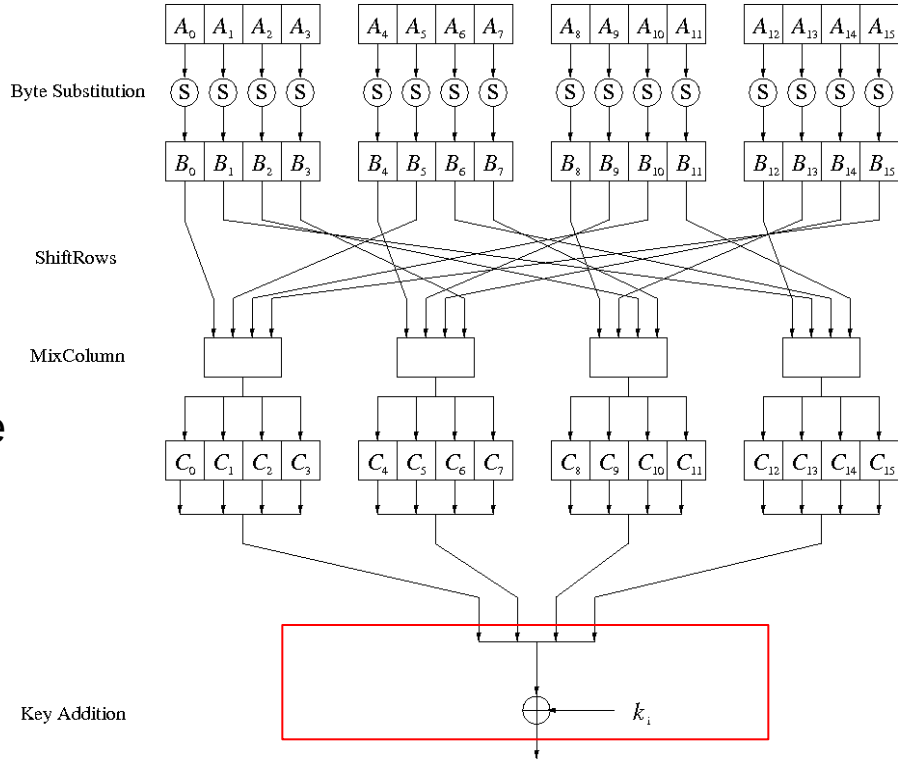
  where 01, 02 and 03 are given in hexadecimal notation

- All arithmetic is done in the Galois field $GF(2^8)$ (for more information see Chapter 4.3 in *Understanding Cryptography*)

Key Addition Layer

The **Key Addition Layer** in the AES algorithm is known as the **AddRoundKey** step. This operation is a crucial part of the AES encryption and decryption process, and it involves the bitwise **XOR** (exclusive OR) of the current state matrix with a **round key** derived from the original encryption key.

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



The **AddRoundKey** step takes the current state matrix (which contains the data being encrypted) and **XORs** it with the corresponding **round key** for that round.

Key Schedule

In AES, **subkeys** are generated from the original key through a process called **key expansion** (or **key schedule**). The expanded keys (subkeys) are used during the encryption rounds to ensure that each round operates on a different portion of the original key.

- 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

The number of subkeys is equal to the number of rounds in AES plus 1 (i.e., **# rounds + 1**)

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

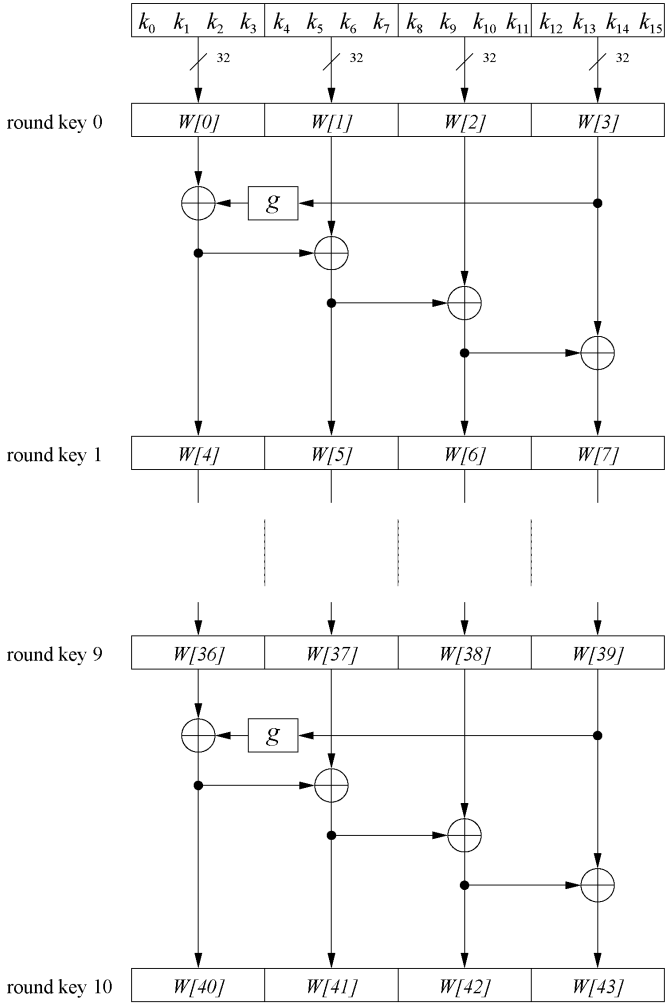
  $\Rightarrow$ # subkeys = # rounds + 1

Key whitening refers to the use of a subkey both **at the input** (before the first round) and **at the output** (after the last round). This strengthens the encryption by directly involving the key in the beginning and at the end.

- 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]…W[3], W[4]…W[7], … , W[40]…W[43]*
- First subkey *W[0]…W[3]* is the original AES key

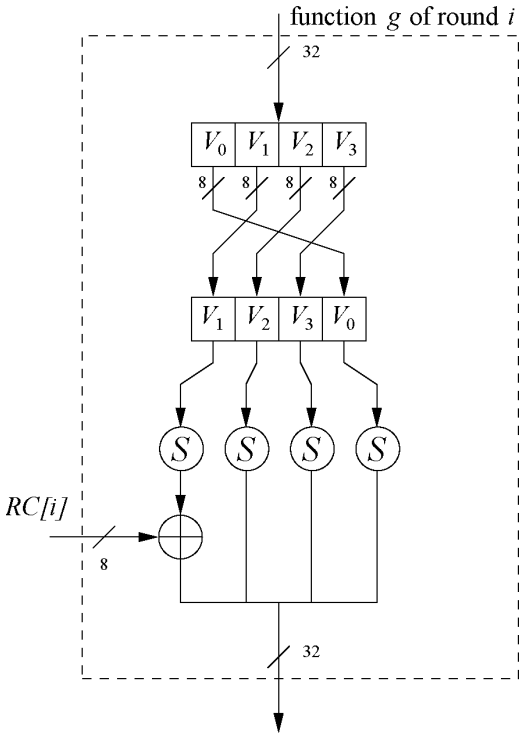
The **key schedule** process generates **word-oriented** subkeys. A **word** in AES refers to **32 bits** (or 4 bytes). The key expansion process produces multiple words to form subkeys, which are applied during the encryption rounds in the **AddRoundKey** step

Key Schedule

- Function *g* rotates its four input bytes and performs a bytewise S-Box substitution
  $\Rightarrow$ nonlinearity

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

$$RC[1] = x^0 = (00000001)_2$$
$$RC[2] = x^1 = (00000010)_2$$
$$RC[3] = x^2 = (00000100)_2$$
$$\ldots$$
$$RC[10] = x^9 = (00110110)_2$$



These coefficients are used only in the key expansion process to ensure that each round key is distinct.

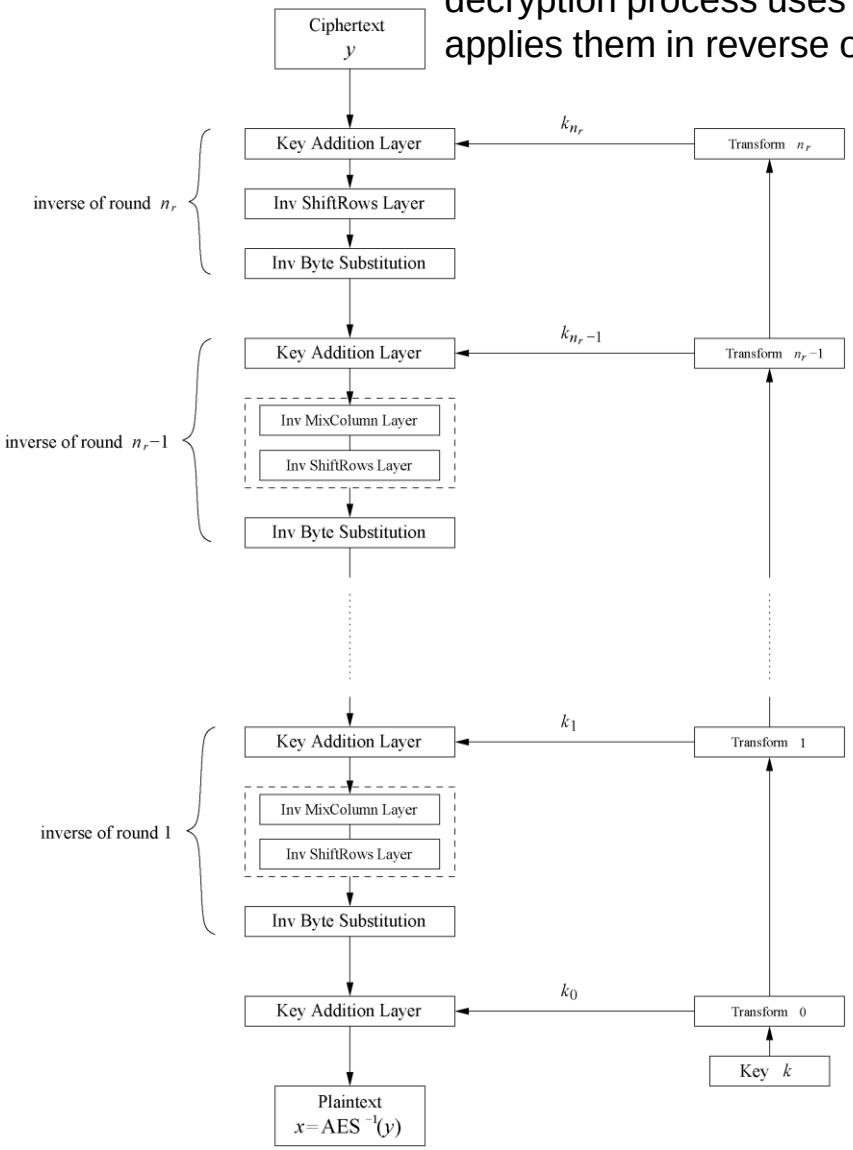
- x^i represents an element in a Galois field
  (again, cf. Chapter 4.3 of *Understanding Cryptography*)

Content of this Chapter

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

Decryption

AES decryption involves performing the reverse operations of the encryption process, but the transformations must be applied in the opposite order. The decryption process uses the same subkeys generated during encryption but applies them in reverse order.



- 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, 0*B, 0D* and 0*E* are given in hexadecimal notation

- Again, all arithmetic is done in the Galois field $GF(2^8)$ (for more information see Chapter 4.3 in *Understanding Cryptography*)

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}	no shift
B_{13}	B_1	B_5	B_9	→ one position right shift
B_{10}	B_{14}	B_2	B_6	→ two positions right shift
B_7	B_{11}	B_{15}	B_3	→ three positions right shift

Decryption

- **Inv Byte Substitution layer**:
  - Since the S-Box is bijective, it is possible to construct an inverse, such that

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

    $\Rightarrow$ 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

Content of this Chapter

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

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

Advantages of AES

The AES algorithm provides several advantages over older algorithms such as the Data Encryption Standard (DES):

•**Security.** AES offers stronger security since it incorporates multiple rounds of encryption, making it harder to break, and harder for threat actors to intercept or steal the encrypted information using brute-force attacks.

•**Cost.** AES is an open source and ubiquitously available solution, making it cost-effective to adopt and implement.

•**Implementation.** AES is a flexible and simple algorithm, making it suitable for both hardware and software implementation.

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.


Thank
You