
بسم الله الرحمن الرحيم

Network Security

Ali Shakiba
Vali-e-Asr University of Rafsanjan
ali.shakiba@vru.ac.ir
www.1ali.ir

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

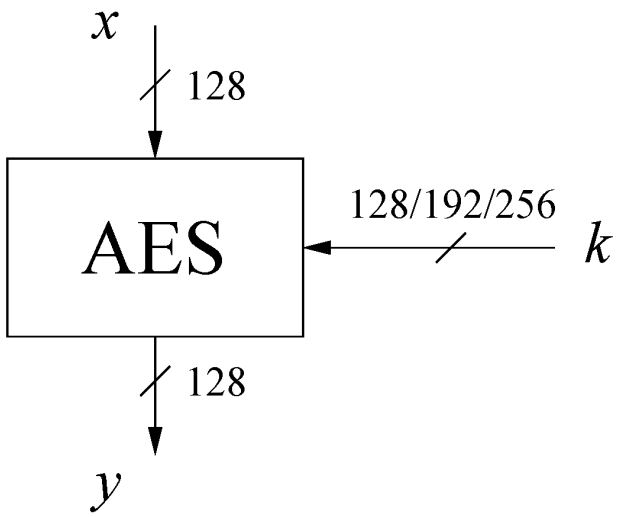
Some Basic Facts

- 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

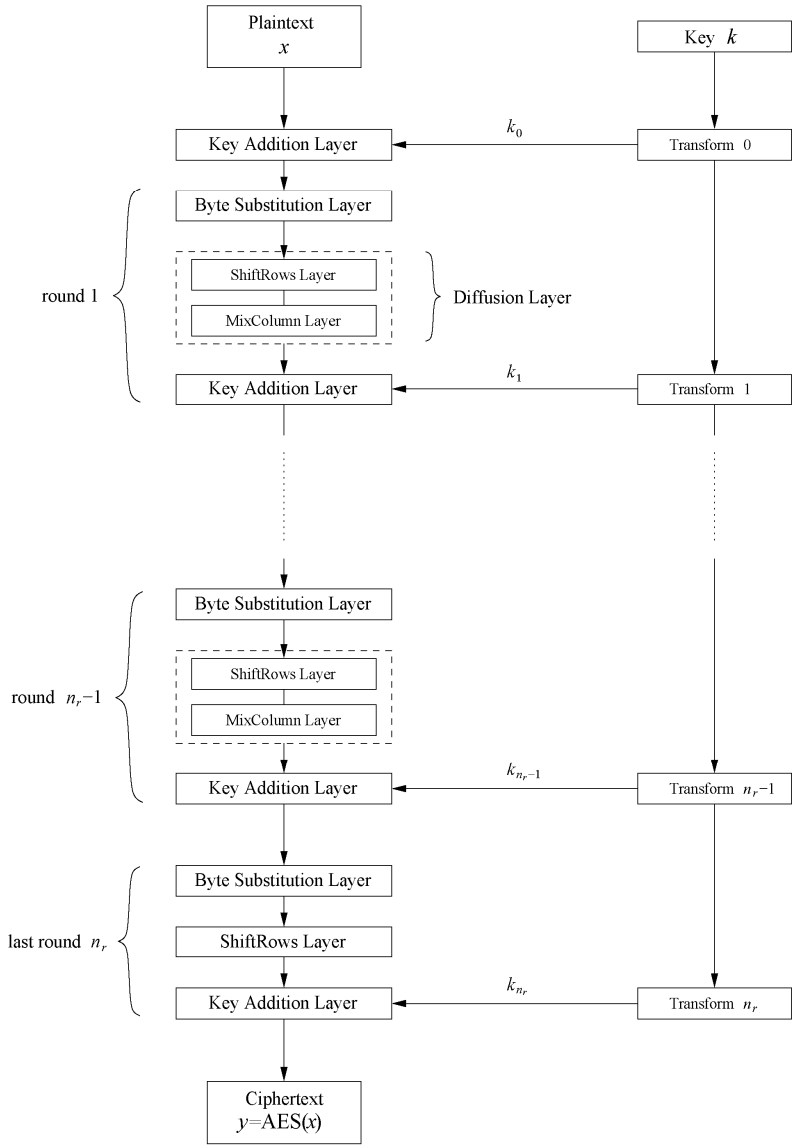


The number of rounds depends on the chosen key length:

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

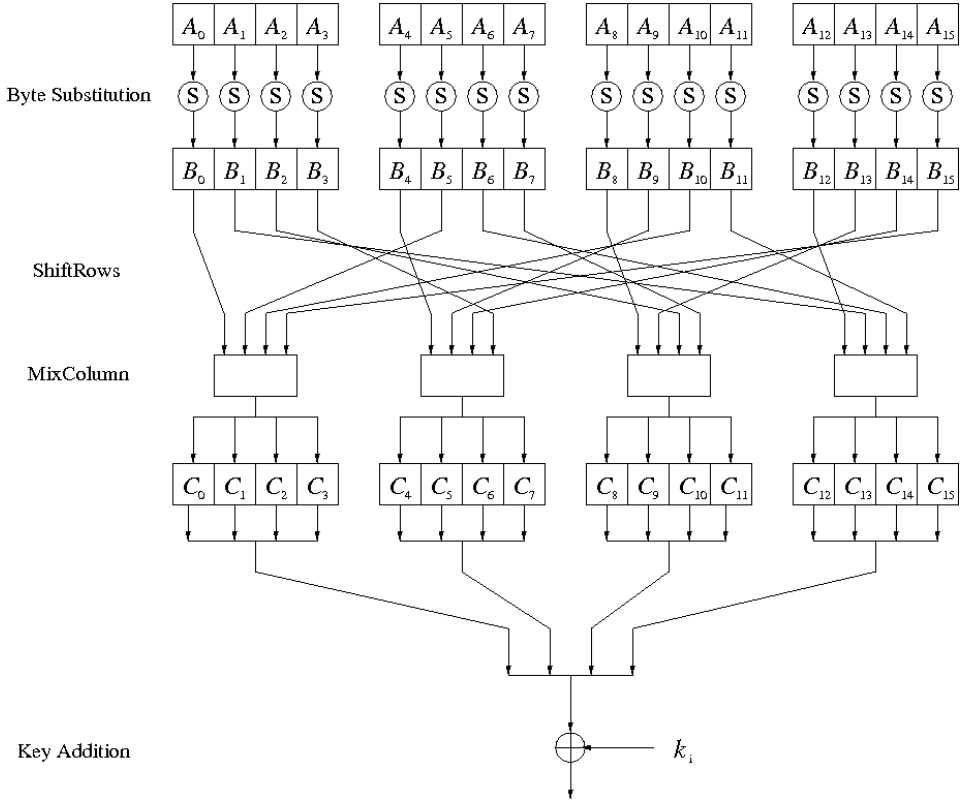
- AES is a byte-oriented cipher
- The state A (i.e., the 128-bit data path) can be arranged in a 4x4 matrix:

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,..., A_{15} denoting the 16-byte input of AES

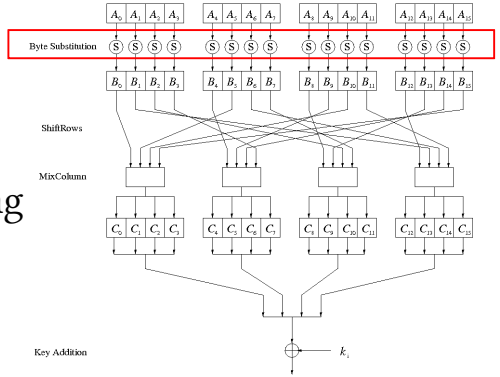
Internal Structure of AES

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



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

Byte Substitution Layer



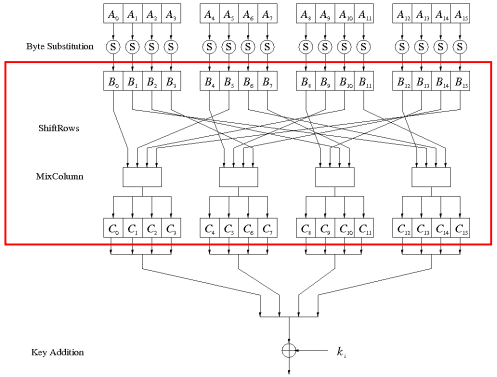
- The Byte Substitution layer consists of 16 **S-Boxes** with the following properties:

  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

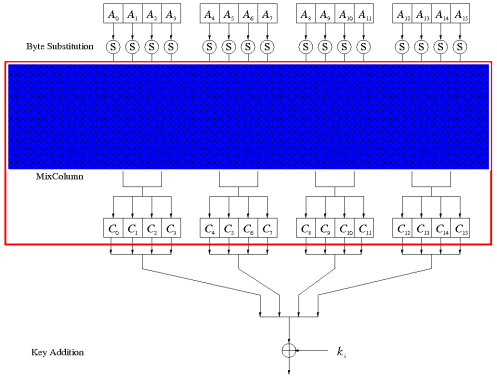
- provides diffusion over all input state bits

- 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.,

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

ShiftRows Sublayer



- Rows of the state matrix are shifted cyclically:

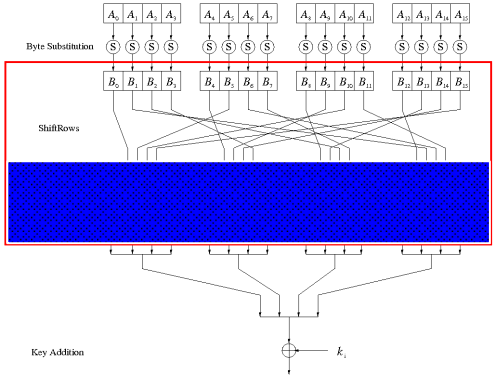
Input matrix

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

MixColumn Sublayer



- Linear transformation which mixes each column of the state matrix

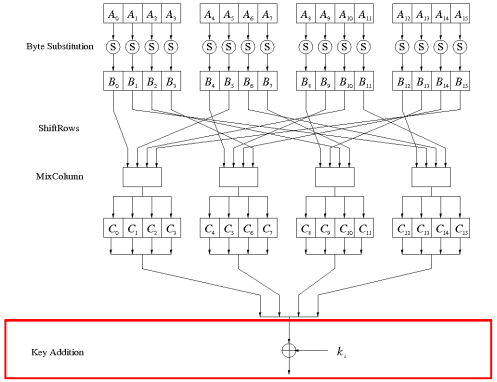
- Each 4-byte column is considered as a vector and multiplied by a fixed 4x4 matrix, e.g.,

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

  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



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

Key Schedule

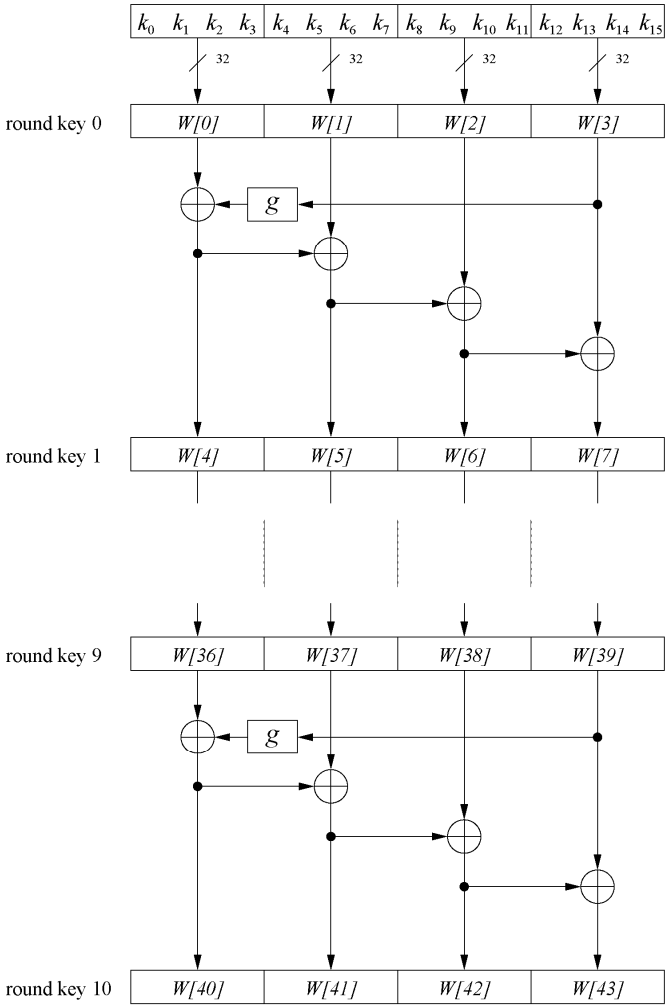
- Subkeys are derived recursively from the original 128/192/256-bit input key
- Each round has 1 subkey, plus 1 subkey at the beginning of AES

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
  $\Rightarrow$ # subkeys = # 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]…W[3], W[4]…W[7], … , W[40]…W[43]*
- First subkey *W[0]…W[3]* is the original AES key

$$i = 1, \ldots, 10, j = 1,2,3$$

$$W[4i] = W[4(i-1)] + g(W[4i-1])$$

$$W[4i+j] = W[4i+j-1] + W[4(i-1)+j]$$

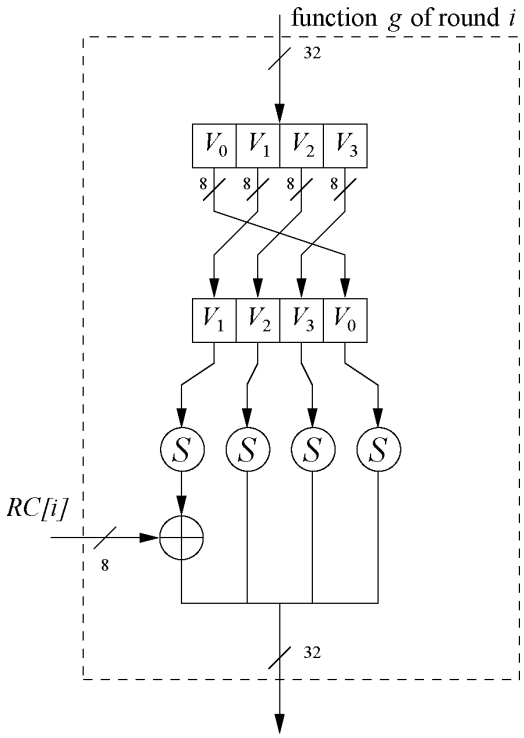
Key Schedule

- 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:

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

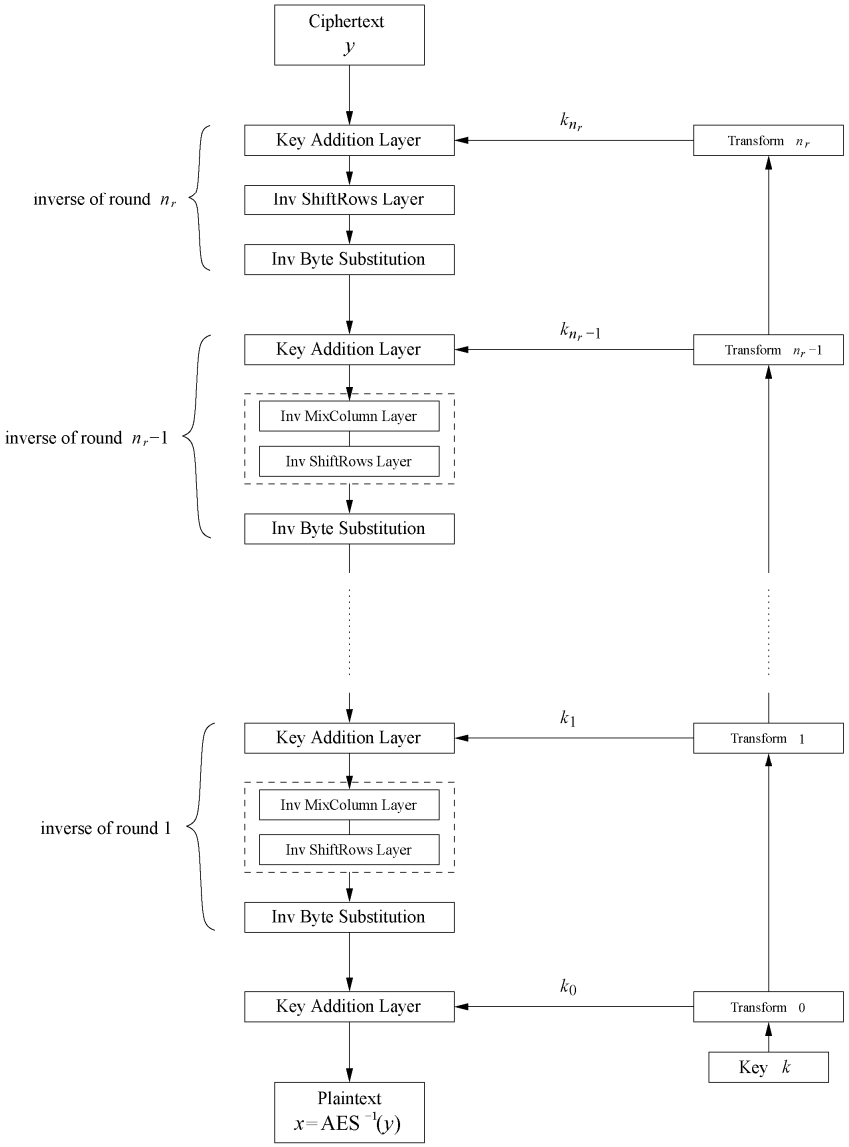
- x^i represents an element in a Galois field
  (again, cf. Chapter 4.3 of *Understanding Cryptography*)



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 is not based on a Feistel network

$\Rightarrow$ All layers must be inverted for decryption:

- MixColumn layer $\longrightarrow$ **Inv MixColumn layer**
- ShiftRows layer$\longrightarrow$ **Inv ShiftRows layer**
- Byte Substitution layer $\longrightarrow$ **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 $GF(2^8)$ (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

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.

Content of this Chapter

- Encryption with Block Ciphers: Modes of Operation
  - Electronic Code Book mode (ECB)
  - Cipher Block Chaining mode (CBC)
  - Output Feedback mode (OFB)
  - Cipher Feedback mode (CFB)
  - Counter mode (CTR)
  - Galois Counter Mode (GCM)
- Exhaustive Key Search Revisited
- Increasing the Security of Block Ciphers

Block Ciphers

- A block cipher is much more than just an encryption algorithm, it can be used ...
  - to build different types of block-based encryption schemes
  - to realize stream ciphers
  - to construct hash functions
  - to make message authentication codes
  - to build key establishment protocols
  - to make a pseudo-random number generator
  - ...

- The security of block ciphers also can be increased by
  - key whitening
  - multiple encryption

Content of this Chapter

- **Encryption with Block Ciphers: Modes of Operation**
  - Electronic Code Book mode (ECB)
  - Cipher Block Chaining mode (CBC)
  - Output Feedback mode (OFB)
  - Cipher Feedback mode (CFB)
  - Counter mode (CTR)
  - Galois Counter Mode (GCM)
- Exhaustive Key Search Revisited
- Increasing the Security of Block Ciphers

Encryption with Block Ciphers

- There are several ways of encrypting long plaintexts, e.g., an e-mail or a computer file, with a block cipher ("modes of operation")
  - Electronic Code Book mode (ECB)
  - Cipher Block Chaining mode (CBC)
  - Output Feedback mode (OFB)
  - Cipher Feedback mode (CFB)
  - Counter mode (CTR)
  - Galois Counter Mode (GCM)

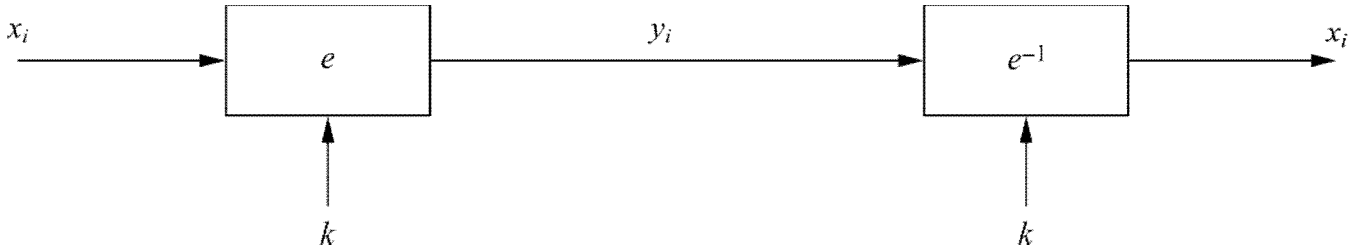
- All of the 6 modes have one goal:
  - In addition to confidentiality, they provide authenticity and integrity:
    - Is the message really coming from the original sender? (authenticity)
    - Was the ciphertext altered during transmission? (integrity)

Content of this Chapter

- Encryption with Block Ciphers: Modes of Operation
  - Electronic Code Book mode (ECB)
  - Cipher Block Chaining mode (CBC)
  - Output Feedback mode (OFB)
  - Cipher Feedback mode (CFB)
  - Counter mode (CTR)
  - Galois Counter Mode (GCM)
- Exhaustive Key Search Revisited
- Increasing the Security of Block Ciphers

Electronic Code Book mode (ECB)

- $e_k(x_i)$ denote the encryption of a b-bit plaintext block x_i with key k
- $e_k^{-1}(y_i)$ denote the decryption of b-bit ciphertext block y_i with key k
- Messages which exceed b bits are partitioned into b-bit blocks
- Each Block is encrypted separately



Encryption: $y_i = e_k(x_i), \quad i \geq 1$

Decryption: $x_i = e_k^{-1}(y_i) = e_k^{-1}(e_k(x_i)), \quad i \geq 1$

ECB: advantages/disadvantages

- Advantages
  - no block synchronization between sender and receiver is required
  - bit errors caused by noisy channels only affect the corresponding block but not succeeding blocks
  - Block cipher operating can be parallelized
    - advantage for high-speed implementations

- Disadvantages
  - ECB encrypts highly deterministically
    - identical plaintexts result in identical ciphertexts
    - an attacker recognizes if the same message has been sent twice
    - plaintext blocks are encrypted independently of previous blocks
      - an attacker may reorder ciphertext blocks which results in valid plaintext

Substitution Attack on ECB

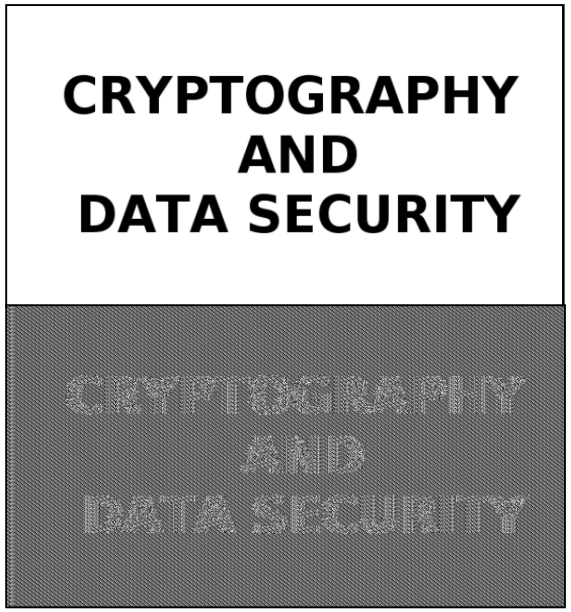
- Once a particular plaintext to ciphertext block mapping $x_i \longrightarrow y_i$ is known, a sequence of ciphertext blocks can easily be manipulated
- Suppose an *electronic bank transfer*

Block #	1	2	3	4	5
	Sending Bank A	Sending Account #	Receiving Bank B	Receiving Account #	Amount $

  - the encryption key between the two banks does not change too frequently
  - The attacker sends $1.00 transfers from his account at bank A to his account at bank B repeatedly
    - He can check for ciphertext blocks that repeat, and he stores blocks 1,3 and 4 of these transfers
  - He now simply replaces block 4 of other transfers with the block 4 that he stored before
    - *all transfers* from some account of bank A to some account of bank B are redirected to go into the attacker's B account!

Example of encrypting bitmaps in ECB mode

- Identical plaintexts are mapped to identical ciphertexts



- Statistical properties in the plaintext are preserved in the ciphertext

Content of this Chapter

- **Encryption with Block Ciphers: Modes of Operation**
  - Electronic Code Book mode (ECB)
  - **Cipher Block Chaining mode (CBC)**
  - Output Feedback mode (OFB)
  - Cipher Feedback mode (CFB)
  - Counter mode (CTR)
  - Galois Counter Mode (GCM)
- Exhaustive Key Search Revisited
- Increasing the Security of Block Ciphers

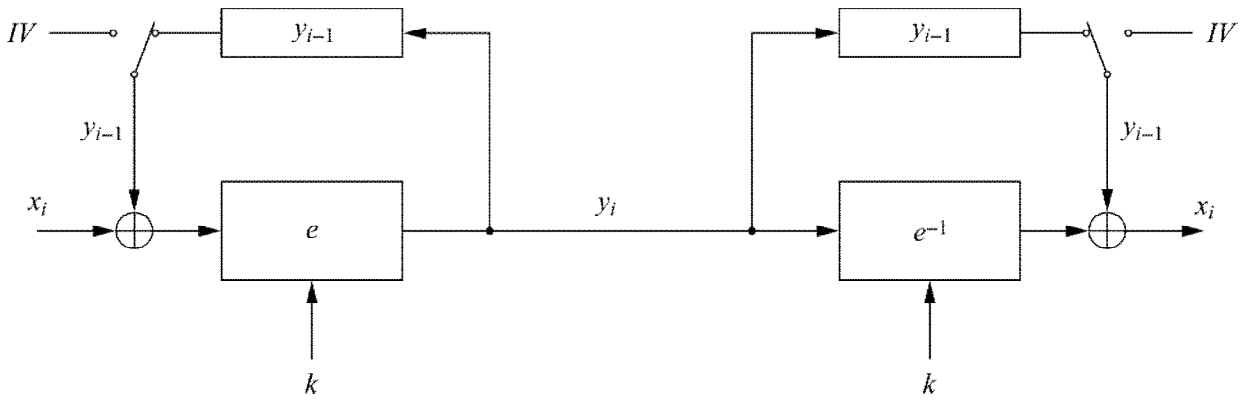
Cipher Block Chaining mode (CBC)

- There are two main ideas behind the CBC mode:
  - The encryption of all blocks are "chained together"
    - ciphertext y_i depends not only on block x_i but on all previous plaintext blocks as well
  - The encryption is randomized by using an initialization vector (IV)

Encryption (first block):	$y_1 = e_k(x_1 \oplus \text{IV})$
Encryption (general block):	$y_i = e_k(x_i \oplus y_{i-1}), \quad i \geq 2$
Decryption (first block):	$x_1 = e_k^{-1}(y_1) \oplus \text{IV}$
Decryption (general block) :	$x_i = e_k^{-1}(y_i) \oplus y_{i-1}, \quad i \geq 2$

Cipher Block Chaining mode (CBC)

- For the first plaintext block x_1 there is no previous ciphertext
  - an IV is added to the first plaintext to make each CBC encryption nondeterministic
  - the first ciphertext y_1 depends on plaintext x_1 and the IV
- The second ciphertext y_2 depends on the IV, x_1 *and* x_2
- The third ciphertext y_3 depends on the IV and x_1, x_2 *and* x_3, and so on



Substitution Attack on CBC

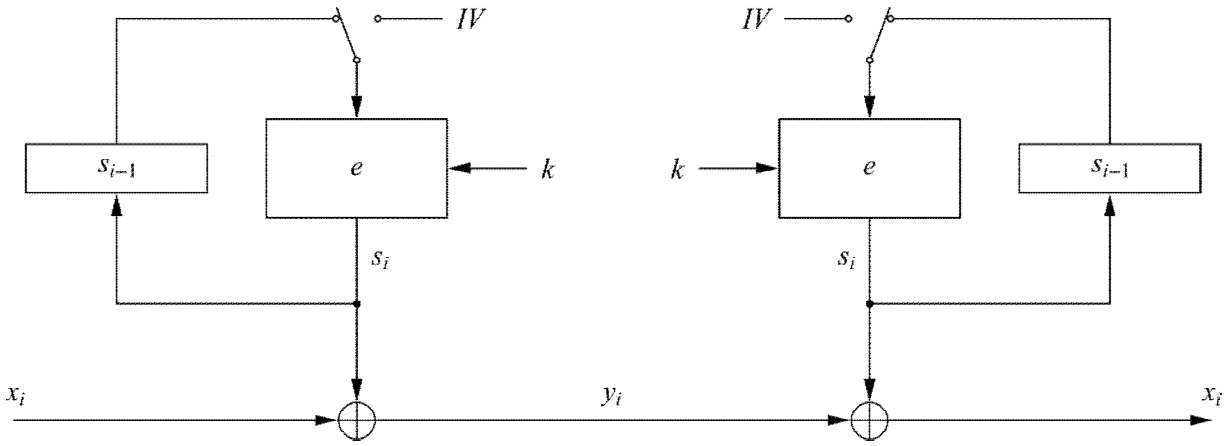
- Suppose the last example (*electronic bank transfer*)
- If the IV is properly chosen for every wire transfer, the attack will not work at all
- If the IV is kept the same for several transfers, the attacker would recognize the transfers from his account at bank A to back B
- If we choose a new IV every time we encrypt, the CBC mode becomes a probabilistic encryption scheme, i.e., two encryptions of the same plaintext look entirely different
- It is not needed to keep the IV *secret*!
- Typically, the IV should be a non-secret nonce (value used only once)

Content of this Chapter

- **Encryption with Block Ciphers: Modes of Operation**
  - Electronic Code Book mode (ECB)
  - Cipher Block Chaining mode (CBC)
  - **Output Feedback mode (OFB)**
  - Cipher Feedback mode (CFB)
  - Counter mode (CTR)
  - Galois Counter Mode (GCM)
- Exhaustive Key Search Revisited
- Increasing the Security of Block Ciphers

Output Feedback mode (OFB)

- It is used to build a *synchronous* **stream cipher** from a block cipher
- The key stream is not generated bitwise but instead in a blockwise fashion
- The output of the cipher gives us key stream bits S_i with which we can encrypt plaintext bits using the XOR operation



Encryption (first block): $s_1 = e_k(\mathrm{IV})$ *and* $y_1 = s_1 \oplus x_1$

Encryption (general block): $s_i = e_k(s_{i-1})$ *and* $y_i = s_i \oplus x_i$, $i \geq 2$

Decryption (first block): $s_1 = e_k(\mathrm{IV})$ *and* $x_1 = s_1 \oplus y_1$

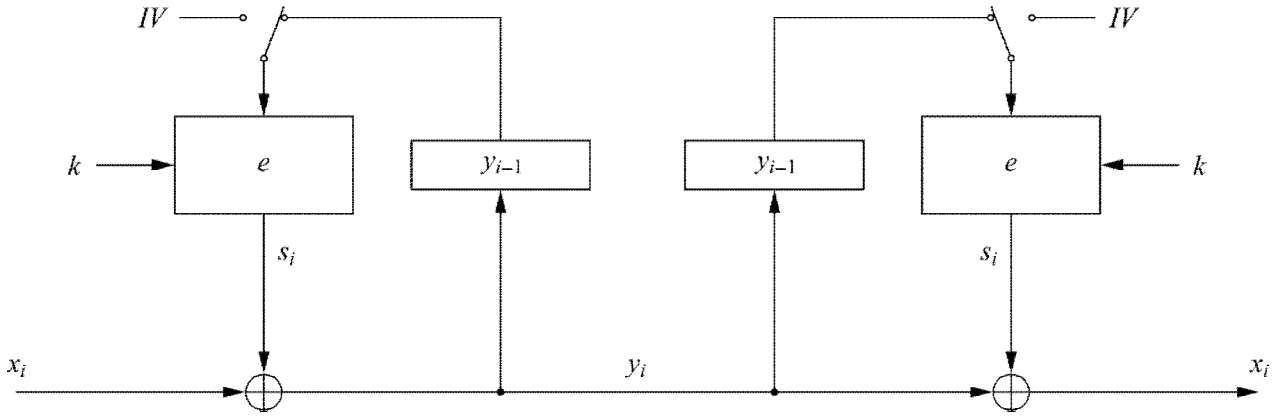
Decryption (general block) : $s_i = e_k(s_{i-1})$ *and* $x_i = s_i \oplus y_i$, $i \geq 2$

Content of this Chapter

- **Encryption with Block Ciphers: Modes of Operation**
  - Electronic Code Book mode (ECB)
  - Cipher Block Chaining mode (CBC)
  - Output Feedback mode (OFB)
  - **Cipher Feedback mode (CFB)**
  - Counter mode (CTR)
  - Galois Counter Mode (GCM)
- Exhaustive Key Search Revisited
- Increasing the Security of Block Ciphers

Cipher Feedback mode (CFB)

- It uses a block cipher as a building block for an asynchronous **stream cipher** (similar to the OFB mode), more accurate name: "Ciphertext Feedback Mode"
- The key stream S_i is generated in a blockwise fashion and is also a function of the ciphertext
- As a result of the use of an IV, the CFB encryption is also nondeterministic



Encryption (first block): $y_1 = e_k(\text{IV}) \oplus x_1$
Encryption (general block): $y_i = e_k(y_{i-1}) \oplus x_i, \quad i \geq 2$
Decryption (first block): $x_1 = e_k(\text{IV}) \oplus y_1$
Decryption (general block) : $x_i = e_k(y_{i-1}) \oplus y_i, \quad i \geq 2$

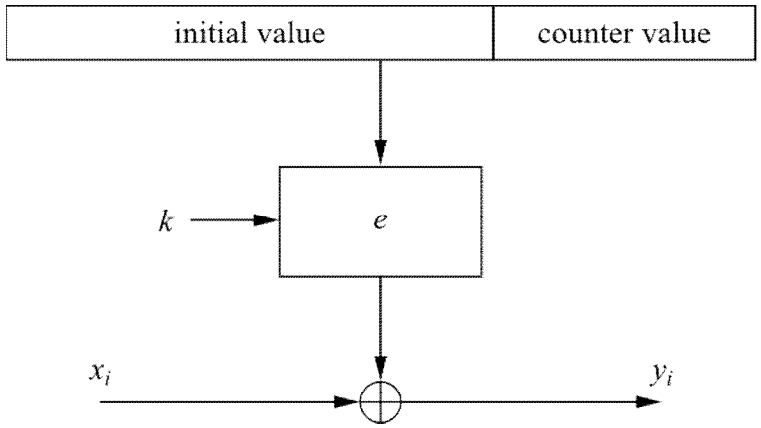
- It can be used in situations where short plaintext blocks are to be encrypted

Content of this Chapter

- **Encryption with Block Ciphers: Modes of Operation**
  - Electronic Code Book mode (ECB)
  - Cipher Block Chaining mode (CBC)
  - Output Feedback mode (OFB)
  - Cipher Feedback mode (CFB)
  - **Counter mode (CTR)**
  - Galois Counter Mode (GCM)
- Exhaustive Key Search Revisited
- Increasing the Security of Block Ciphers

Counter mode (CTR)

- It uses a block cipher as a **stream cipher** (like the OFB and CFB modes)
- The key stream is computed in a blockwise fashion
- The input to the block cipher is a counter which assumes a different value every time the block cipher computes a new key stream block



- Unlike CFB and OFB modes, the CTR mode can be parallelized since the 2nd encryption can begin before the 1st one has finished
  - Desirable for high-speed implementations, e.g., in network routers

$$\textbf{\textit{Encryption}}: \quad y_i = e_k(\text{IV} \,||\, \text{CTR}_i) \oplus x_i, \quad i \geq 1$$
$$\textbf{\textit{Decryption}}: \quad x_i = e_k(\text{IV} \,||\, \text{CTR}_i) \oplus y_i, \quad i \geq 1$$

Content of this Chapter

- **Encryption with Block Ciphers: Modes of Operation**
  - Electronic Code Book mode (ECB)
  - Cipher Block Chaining mode (CBC)
  - Output Feedback mode (OFB)
  - Cipher Feedback mode (CFB)
  - Counter mode (CTR)
  - **Galois Counter Mode (GCM)**
- Exhaustive Key Search Revisited
- Increasing the Security of Block Ciphers

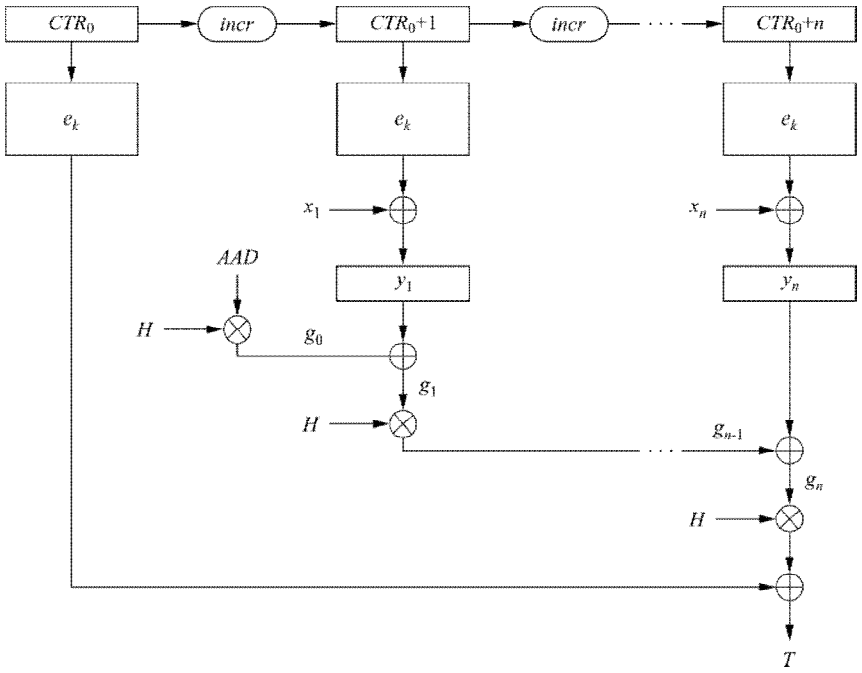
Galois Counter Mode (GCM)

- It also computes a *message authentication code* (MAC), i.e., a cryptographic checksum is computed for a message (for more information see Chapter 12 in *Understanding Cryptography*)
- By making use of GCM, two additional services are provided:
  - Message Authentication
    - the receiver can make sure that the message was really created by the original sender
  - Message Integrity
    - the receiver can make sure that nobody tampered with the ciphertext during transmission

Galois Counter Mode (GCM)

- For encryption
  - An initial counter is derived from an IV and a serial number
  - The initial counter value is incremented then encrypted and XORed with the first plaintext block
  - For subsequent plaintexts, the counter is incremented and then encrypted
- For authentication
  - A chained Galois field multiplication is performed (for more information Galois field see Chapter 4.3 in *Understanding Cryptography*)
  - For every plaintext an intermediate authentication parameter g_i is derived
    - g_i is computed as the XOR of the current ciphertext and the last g_{i-1}, and multiplied by the constant H
      - H is generated by encryption of the zero input with the block cipher
  - All multiplications are in the 128-bit Galois field $\mathrm{GF}(2^{128})$

Galois Counter Mode (GCM)



Encryption:

a. Derive a counter value CTR_0 from the IV and compute $\mathrm{CTR}_1 = \mathrm{CTR}_0 + 1$

b. Compute ciphertext: $y_i = e_k(\mathrm{CTR}_i) \oplus x_i, \quad i \geq 1$

Authentication:

a. Generate authentication subkey $H = e_k(0)$

b. Compute $g_0 = AAD \times H$ (Galois field multiplication)

c. Compute $g_i = (g_{i-1} \oplus y_i) \times H, \; 1 \leq i \leq n$ (Galois field multiplication)

d. Final authentication tag: $T = (g_n \times H) \oplus e_k(\mathrm{CTR}_0)$

Content of this Chapter

- Encryption with Block Ciphers: Modes of Operation
- **Exhaustive Key Search Revisited**
- Increasing the Security of Block Ciphers