ELTE
EÖTVÖS LORÁND
UNIVERSITY

CRYPTOGRAPHY AND SECURITY

Practice

IP-18FKVKRBG

Lecture 7

2022-2023-2

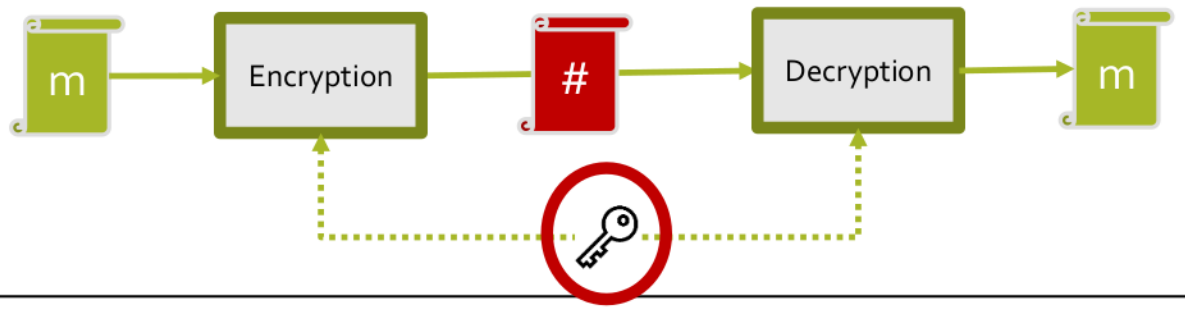
Symmetric Cryptography

Symmetric Cryptography

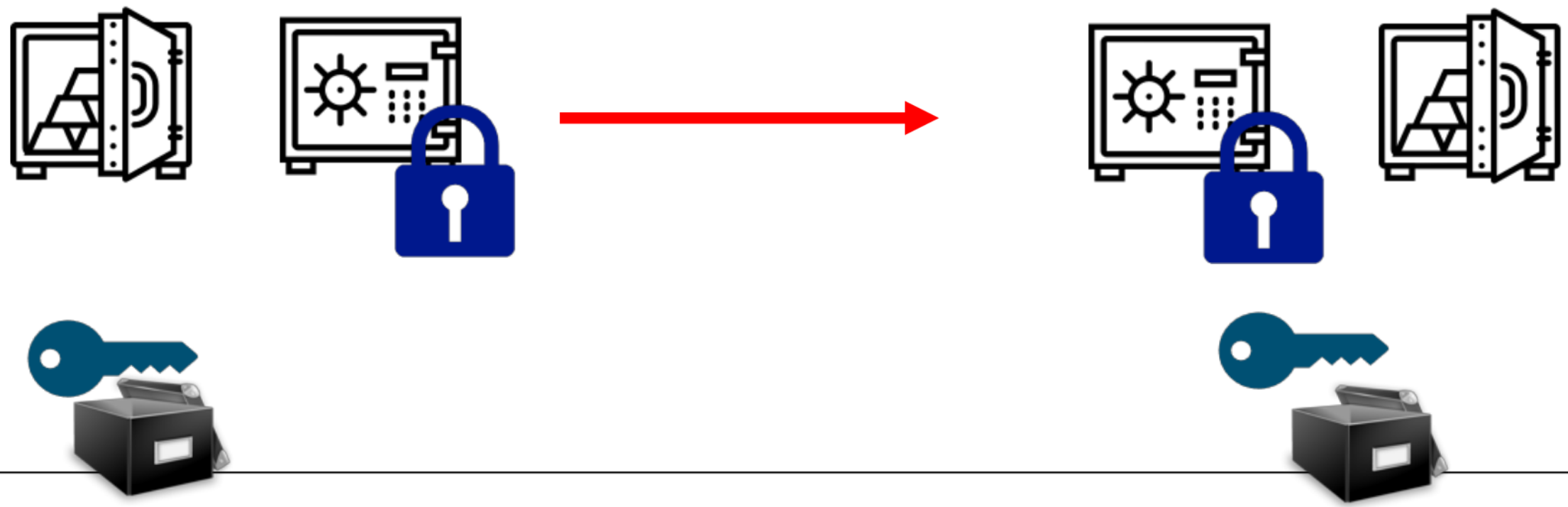
- **Symmetric-key encryption** is a type of encryption where only one key (a secret key) is used to both encrypt and decrypt electronic information.

- Anyone who knows the secret key can decrypt the secret message.

- With symmetric-key encryption, the encryption key can be calculated from the decryption key, and vice versa.

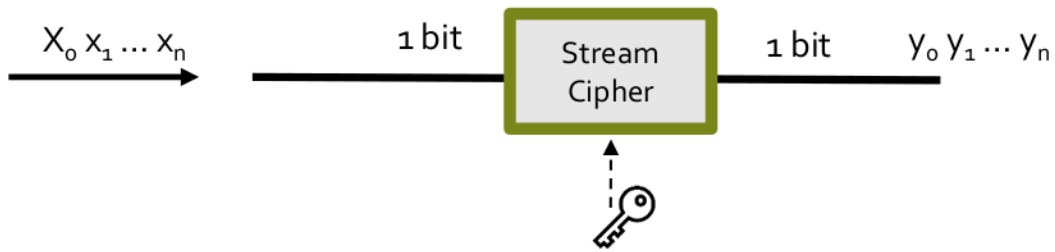


Symmetric Cryptography I

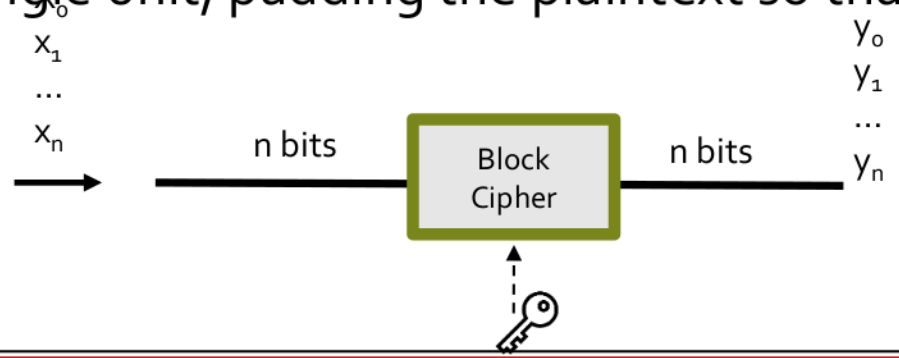


Symmetric Cryptography II

- Stream Cipher: Encrypts one bit at time



- Block Cipher: Encrypts a set of bits (i.e. a block of data, e.g. 64 or 128 bits) at time, encrypt them as a single unit, padding the plaintext so that it is a multiple of the block size.

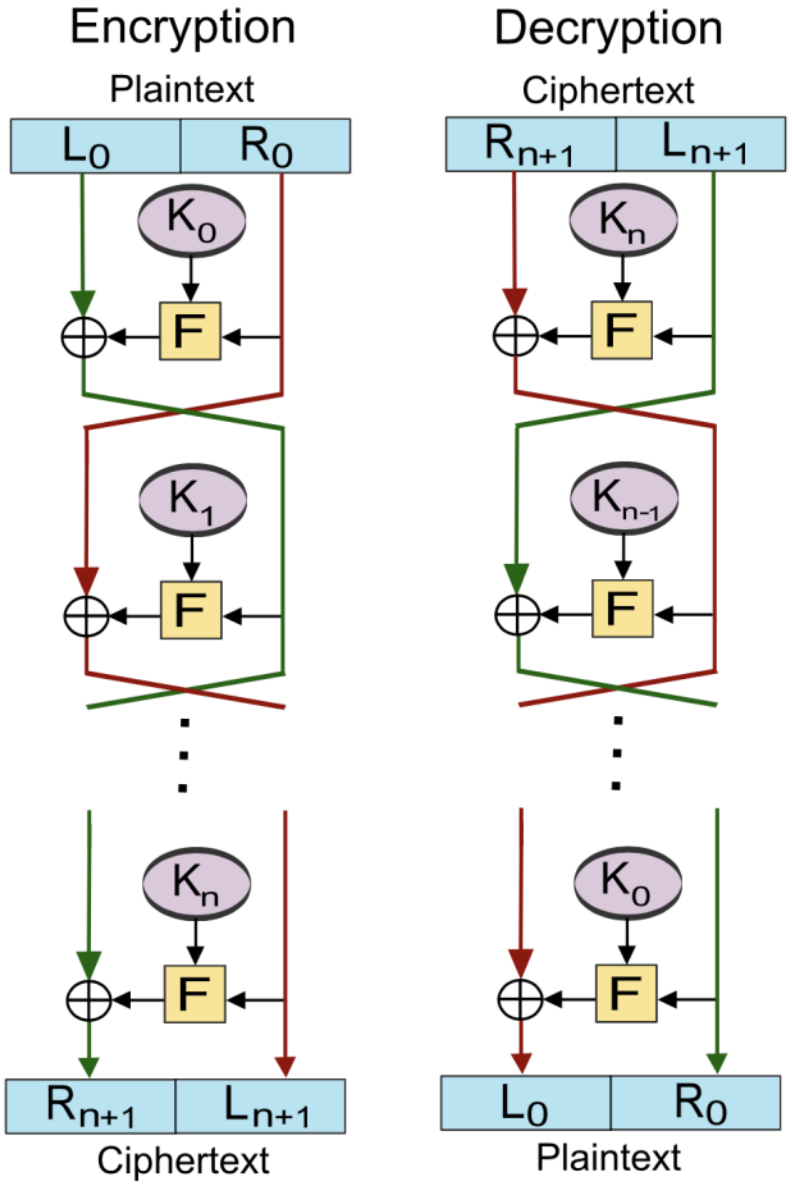


Data Encryption System (DES)

Feistel-network

The entire operation is guaranteed to be invertible.

that is, encrypted data can be decrypted, **even if the round function is not itself invertible**.



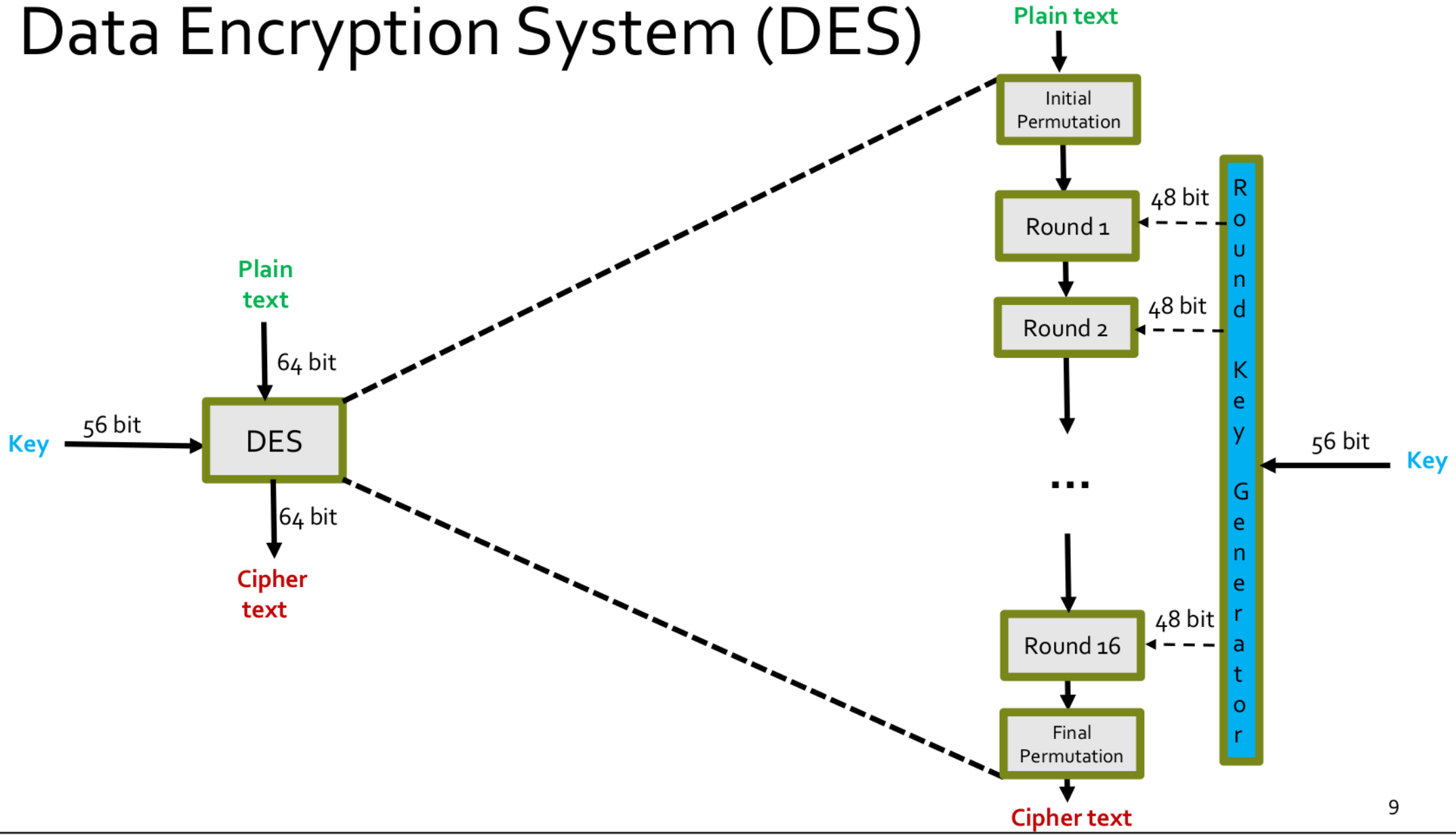
Data Encryption System (DES)

- based on an earlier design by Horst Feistel.
- In 1976, DES is approved as a standard, 1rst published in 1977

- Encrypts **blocks** of size **64 bits**.
- Uses a **key** of size **56 bits**.
- The same (56-bits) key is used both for encryption and decryption.
- Includes 16 rounds. Each of them performs the same set of operations.
- A 48-bits subkey is generated for each round. (Derived from the main key)

- Today DES considered as an insecure algorithm

Data Encryption System (DES)



Some of the Attacks on DES

- In 1977, Diffie and Hellman proposed a machine costing an estimated US$20 million which could find a DES key in a single day.
- In 1991, Biham and Shamir proposed differential cryptanalysis attack that required 247 chosen ciphertexts.
- In 1997, distributed.net could break DES key in 3 months (Costs 250k$)
- In 1998, Deep Crack breaks a DES key in 56 hours
- In 1999, Deep Crack along with distributed.net could break DES key in 22 hours and 15 minutes
- In 2006, COPACOBANA could break DES key in 7 days (Costs 10k$)
- In 2016, using hashcat could recover a key in an average of under 2 days

Alternatives to DES

Algorithm	Input/ Output length	Notes
AES (Rijndael)	128	Standard replacement to DES
Mars	128	AES Finalist
RC6	128	
Serpent	128	
Twofish	128	

Algorithm	Input/ Output length
Triple DES	64
IDEA	64

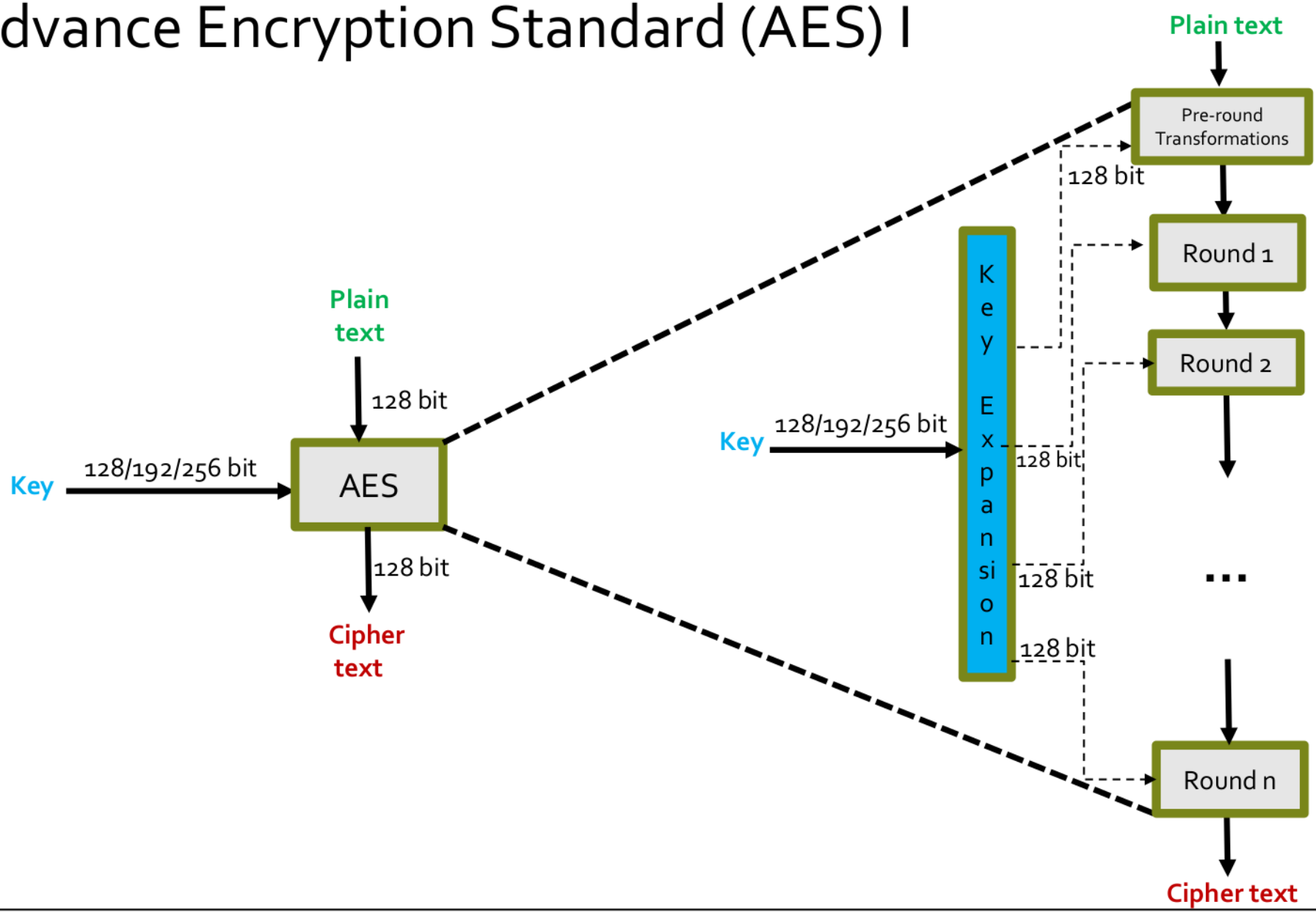
Advance Encryption Standard (AES)

Advance Encryption Standard (AES)

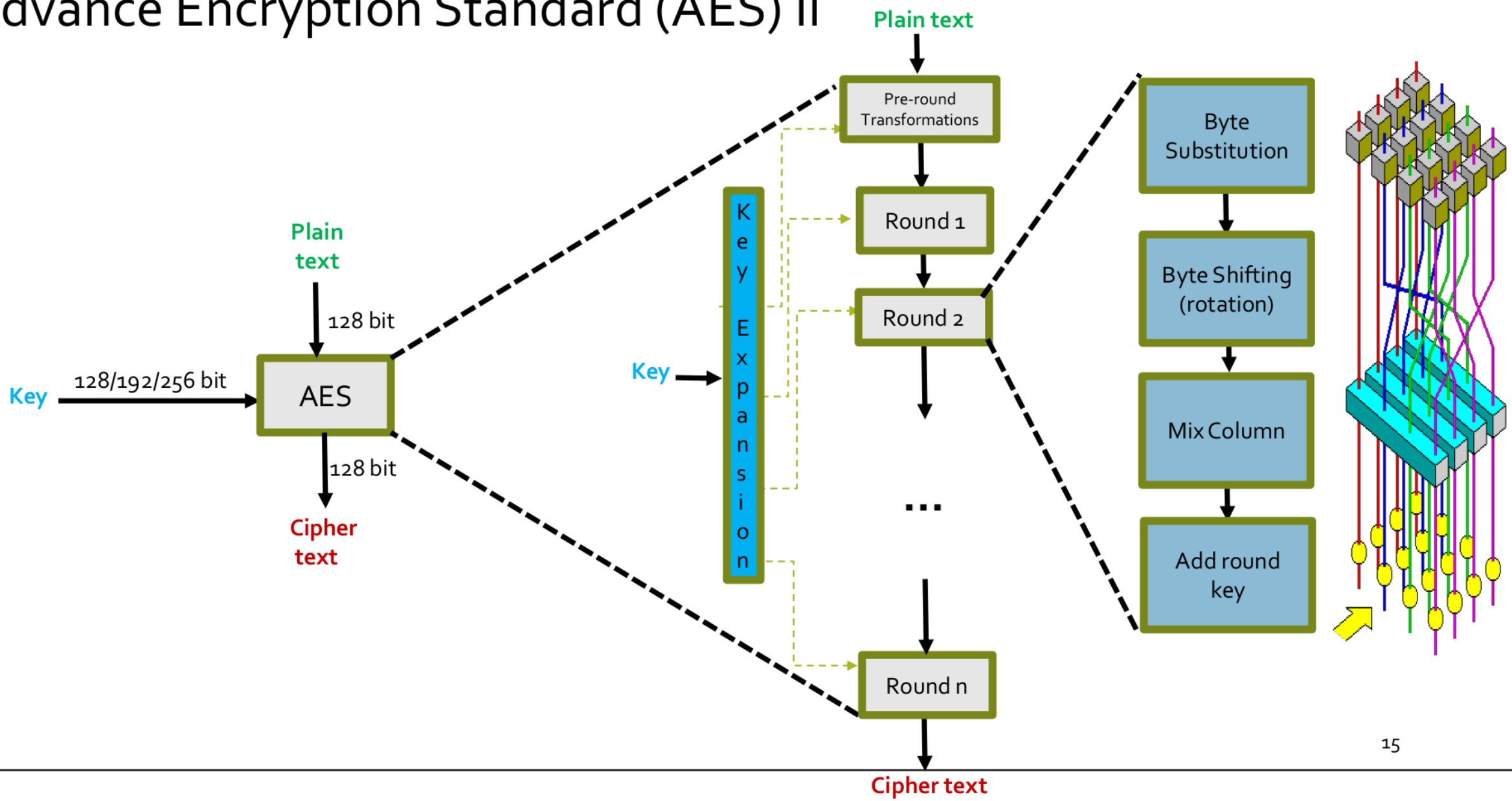
- **Rijndael** developed by Vincent Rijmen and Joan Daemen
- First published in 1998
- The winner among the 5 AES finalists announced in 1999
- Became effective in 2002
- The number of rounds (n) depends on the length of the key:

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

Advance Encryption Standard (AES) I



Advance Encryption Standard (AES) II



Block cipher modes (for DES, AES, …)

- Confidentiality only modes:
  - ECB mode: Electronic Code Book mode
  - CBC mode: Cipher Block Chaining mode
  - CFB mode: Cipher FeedBack mode
  - OFB mode: Output FeedBack mode
  - CTR mode: Counter mode
- Authenticated encryption with additional data (AEAD) modes:
  - Galois/counter mode (GCM)
  - Counter with cipher block chaining message authentication code (CCM)

Cipher Block Chaining (CBC)

- The plaintext is broken into blocks, based on the block size of the used cipher algorithm.
- Each block is XORed (chained) with the ciphertext of the previous block before encryption:

$$C_i = E_K(C_{i-1} \oplus P_i)$$

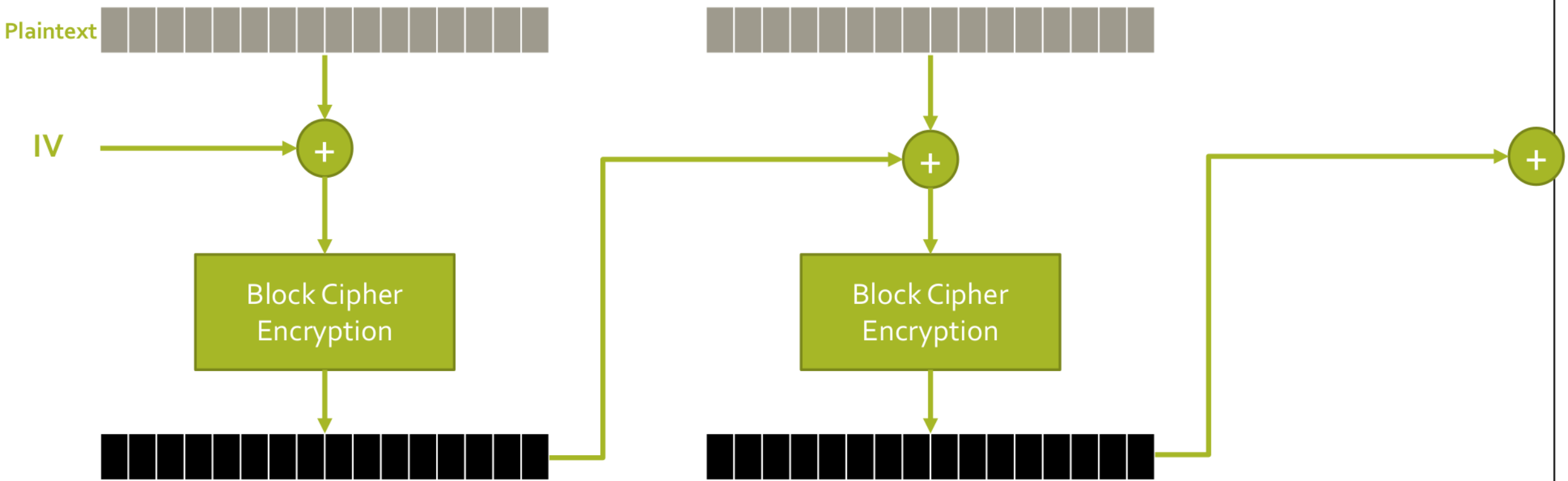
- IV is used to start initiate the process

$$C_1 = E_K(IV \oplus P_1)$$

- Decryption is done as:

$$P_i = C_{i-1} \oplus D_K(C_i)$$

CBC Mode of Encryption



AES (CBC Mode)

Padding

```
import secrets
from Crypto.Cipher import AES

BLOCK_SIZE = 16  # Bytes
pad = lambda s: s + (BLOCK_SIZE - len(s) % BLOCK_SIZE) * \
                chr(BLOCK_SIZE - len(s) % BLOCK_SIZE)
unpad = lambda s: s[:-ord(s[len(s) - 1:])]
```

Encryption

```
    iv =  secrets.token_bytes(16)
    Enc = AES.new(key, AES.MODE_CBC, iv)
    data = pad(plaintext).encode()
    ciphertext = Enc.encrypt(data)
    ciphertext_hex = iv.hex() + ciphertext.hex()
```

Decryption

```
    iv = bytes.fromhex(ciphertext[:32])
    ciphertext = bytes.fromhex(ciphertext[32:])
    Dec = AES.new(key, AES.MODE_CBC, iv)
    pt = Dec.decrypt(ciphertext)
```