Advanced Encryption Standard (AES)

Gianluca Dini
Dept. of Ingegneria dell'Informazione
University of Pisa
gianluca.dini@unipi.it
Version: 2021-03-22

1

AES history

- **1997**: NIST publishes request for proposal
- **1998**: Fifteen proposals
- **1999**: NIST chooses five finalists
  - Mars, RC6, Rijndel, Serpent, Twofish
- **2000**: NIST choses Rijndael as AES
  - Key sizes: 128, 192, 256
    - the longer, the more secure but the slower
  - Block size: 128 bits
- **2003**: NSA allows AES in classified documents
  - Level SECRET: all key lengths
  - Level TOP SECRET: k = 256, 512
  - Never happened before for a public algorithm
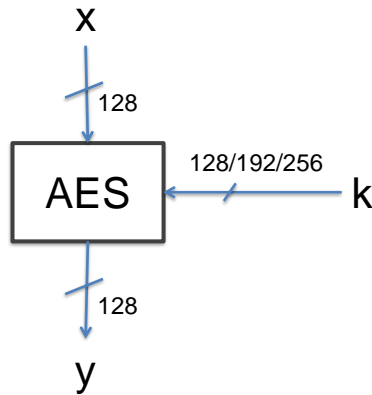
mar. '24   AES   2

2

Overview

x

128

AES ← 128/192/256 ← k

128

y

Key lenght	#rounds
128	10
192	12
256	14

mar. '24 AES 3

3

Introduction

- AES
  - Has rounds
  - Does not have a Feistel network structure
  - Encrypts an entire block in each round
    - DES encrypts half a block ➔ #round$_{AES}$ < #round$_{DES}$
  - Data path is called «state» (4-by-4-byte matrix)
    - 4 bytes of the ciphertext fill the first column,…

mar. '24 AES 4

4

Round and layers

- Every round but the last has four layers
- Layers
  - Key addition layer
  - Byte substitution layer (S-box) - Confusion, Non-linear
  - Diffusion layer is composed of two linear (sub-)layers)
    - *ShiftRows* – permute data byte-wise
    - *MixColumn* – Mix blocks of four bytes (matrix operation)
- Galois fields mathematical setting
  - S-box, MixColumn

Mathematical setting

- Galois field GF(2^8)
  - Operations in S-box and MixColumn are performed in this field
  - Elements of GF(2^m) can be represented as a polynomials of degree $m - 1$ with parameters in GF(2)
    - An element of GF(2^8) represents one byte
      - $A = a_7 x^7 + \dots + a_1 x + a_0$ with $a_i \in$ GF(2) = {0, 1}
      - $A = (a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0)$
    - We cannot use integer arithmetic
    - We must use polynomial arithmetic

Mathematical setting

- Polynomial arithmetic
  - Addition, subtraction
  - Multiplication
    - Core operation of MixColumn
    - Reduction, *irreducible polynomial* (rough equivalent of prime number)
      - A(x) $\times$ B(x) $\equiv$ C(x) mod P(x), with P(x) irreducible polynomial of degree m
      - AES: P(x) = $x^8 + x^4 + x^3 + x^1 + 1$

7

Mathematical setting

- Polynomial arithmetic
  - Division
    - Core operation of Byte Substitution (S-boxes)
    - A(x)·A(x)$^{-1}$ $\equiv$ 1 mod P(x)
    - In small fields (smaller than 2^{16} elements), inverse can be precomputed by lookup tables

8

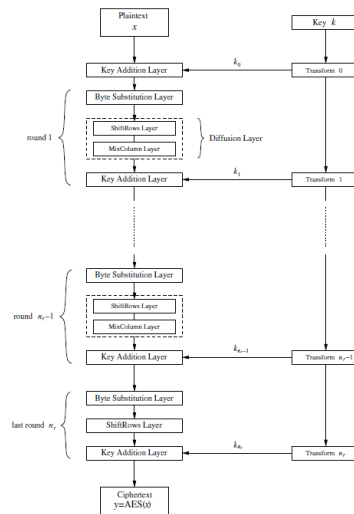# AES encryption block diagram

9

AES is a Subs-Perms network
(not a Feistel network)

10

AES-128 schematics

10 rounds

4

4 input ⊕ | (1) ByteSub (2) ShiftRow (3) MixColumn | ⊕ | (1) ByteSub (2) ShiftRow (3) MixColumn | ⊕ • • • ⊕ | (1) ByteSub (2) ShiftRow |

invertible

k_0 k_1 k_2 k_9

⊕

key

k_{10}

4 output

key expansion:

16 bytes   16 bytes →176 bytes   4
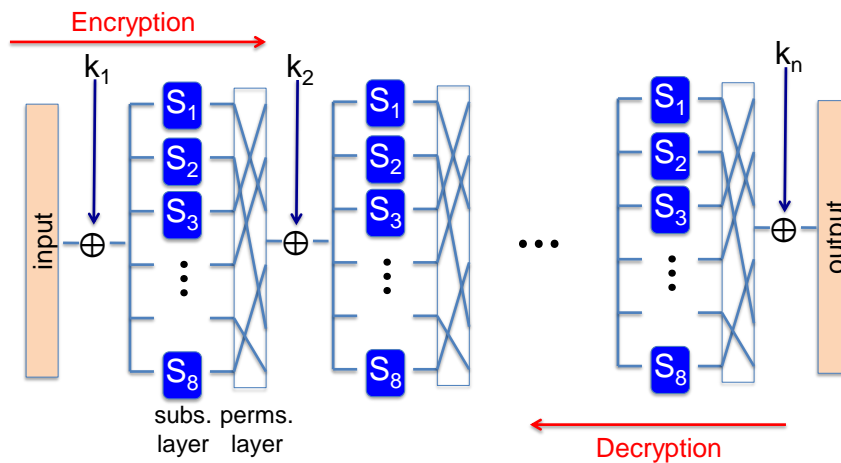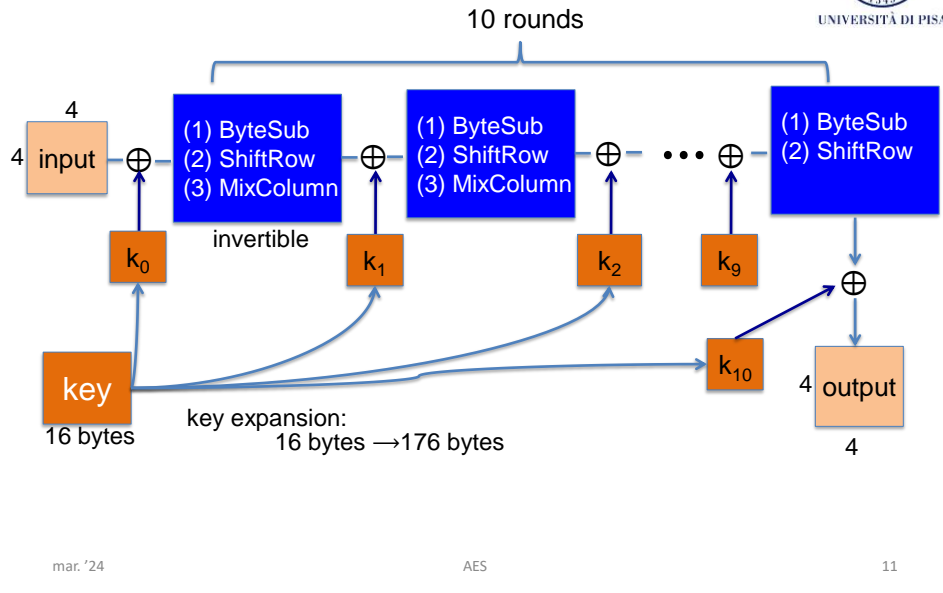
mar. '24   AES   11

11

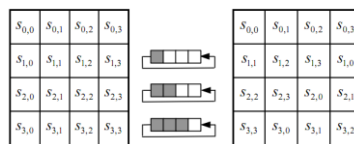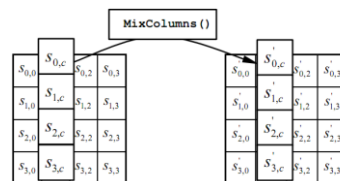# The round function

- **ByteSub**: a 1-byte S-box (256-byte table)
  - Easily computable

- **ShiftRows:**

MixColumns()

- **MixColumns:**
  (linear transformations)

diffusion

mar. '24   AES   12

12

6

3/13/2024

AES Security

- There is currently no analytical attack against AES known to be more efficient than brute force attack

- For more information about AES security see AES Lounge
  - ECRYPT Network of Excellence (FP6)
  - https://www.iaik.tugraz.at/content/research/krypto/aes/

mar. '24   AES   13

13

AES security - best known attacks

- Best key recovery attack
  - Four times better than exhaustive key search
  - 128-bit key → 126-bit key

- "Related key" attack in AES-256
  - Given 2^{99} pt-ct pairs from four related keys in AES-256, we can recover keys in 2^{99} ($\ll 2^{256}$)
    - Very large data-/time-complexity
    - Randomly generated keys cannot be related

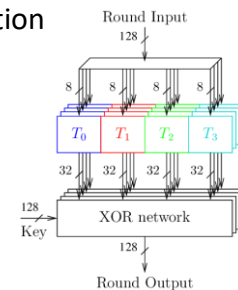mar. '24   AES   14

14

AES Performance (1/2)

- Software implementation
  - Direct implementation is well-suited for 8-bit processors (e.g., smartcard)
    - Processing 1-byte per instruction
  - For 32-/64-bit architecture, T-box optimization
    - Merge all the round functions into one look-up table (but key addition)
      - 4 tables (1 per byte) of 256 entries; each entry is 32 bit
      - 1 round, 16 lookups
  - Few hundreds Mbit/s

mar. '24   AES   15

15

AES Implementation (2/2)

- Hardware implementation
  - AES requires more HW resources than DES
    - High throughput implementation in ASIC/FPGA
    - Ten Gigabit/s
  - Block cipher is extremely fast compared to
    - Asymmetric algorithms
    - Compression algorithms
    - Signal processing algorithms
  - For more information see AES Lounge

mar. '24   AES   16

16

Code size/performance tradeoff

	Code size	Performance
Pre-compute round functions (24KB or 4 KB)	Largest	Fastest (table lookups and xors)
Pre-compute S-box only (256 bytes)	Smaller	Slower
No pre-computation	Smallest	Slowest

mar. '24   AES   17

17

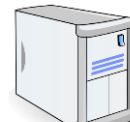# Example: Javascript AES
(Stanford Javascript Crypto Library)

AES in the browser:

AES library (6.4KB)

no pre-computed tables

Prior to encryption:
    pre-compute tables

Then encrypt using tables

http://crypto.stanford.edu/sjcl/

mar. '24   AES   18

18

AES in hardware

- AES instructions in Intel Westmere
  - aesenc, aesenclast: do one round of AES
    - 128-bit registers: xmm1 = state, xmm2 = round key
    - aesenc xmm1, xmm2 puts result in xmm1
  - aeskeygenassist performs key expansion
  - Implement AES in ten instructions
    - 9x aesenc + aesenclast
  - Claim 14x speed-up over OpenSSL on the same hw
- Similar instructions for AMD Bulldozer

mar. '24   AES   19

19