Advanced Encryption Standard (AES)

Gianluca Dini
Dept. of Ingegneria dell'Informazione
University of Pisa
gianluca.dini@unipi.it
Version: 11/03/2025

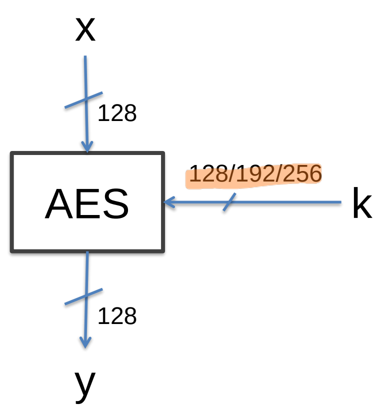
1

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 (with already quantum computers in mind)
    - the longer, the more secure but the slower
  - Block size: 128 bits Discourage dictionary attacks and to implement hash funct.
- **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 AES considered so secure to be used in mil. defence

2

Overview



Key lenght	#rounds
128	10
192	12
256	14

PRODUCT CYPHER
• Each round introduces confusion and diffusion

Introduction

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

plaintext block moves through rounds and states are represented by matrix. Each element of the matrix can be considered a polinomyal

Round and layers

- Every round but the last has four layers
- Layers
  1. Key addition layer Where key is mixed with pt (confusion)
  2. Byte substitution layer (S-box) - Confusion, Non-linear
  3. Diffusion layer is composed of two linear (sub-)layers
     a. *ShiftRows* – permute data byte-wise Rotate row of matrix
     b. *MixColumn* – Mix blocks of four bytes (matrix operation) Compute a sort of linear comb. of 2 matrixes (sort of scalar product)
- Galois fields mathematical setting
  – S-box, MixColumn

Math setting is Galois fields

mar. '25 AES 5

5

Mathematical setting

- Galois field GF(2^8) "Trick" to perform operations on bytes
  – 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)
    Each bit can be considered as coefficient of a polynomial
    - An element of GF(2^8) represents one byte
      – $A = a_7x^7 + \ldots + a_1x + a_0$ 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

mar. '25 AES 6

6

Mathematical setting

- Polynomial arithmetic
  - Addition, subtraction you perform addition coefficient wise and they are XOR
  - Multiplication
    - Core operation of **MixColumn**
    - Reduction, *irreducible polynomial* (rough equivalent of prime number)
      - $A(x) \times B(x) \equiv C(x) \bmod P(x)$, with $P(x)$ irreducible polynomial of degree $\boldsymbol{m}$
      - AES: $P(x) = x^8 + x^4 + x^3 + x^1 + 1$

modulus is not a number, but a polynomial. Cannot be decomposed in other polynomials.

This can be performed in an efficient way.

mar. '25 AES 7

7

Mathematical setting

- Polynomial arithmetic
  - Division
    - Core operation of **Byte Substitution** (S-boxes)
    - $A(x) \cdot A(x)^{-1} \equiv 1 \bmod P(x)$ We define the inverse of a polynomial
    - In small fields (smaller than 2^{16} elements), inverse can be precomputed by lookup tables

our case, 2^8

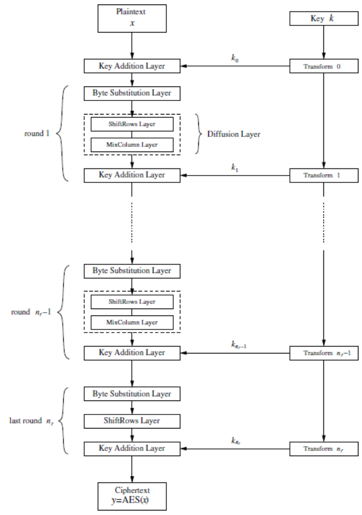
With enough storage you get more efficient

Otherwise you perform less computations so less space used

mar. '25 AES 8

8

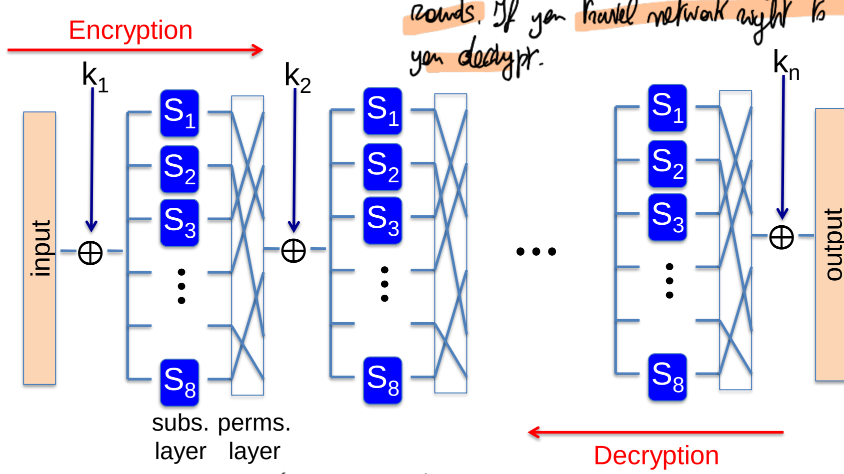
AES encryption block diagram



AES is a Subs-Perms network

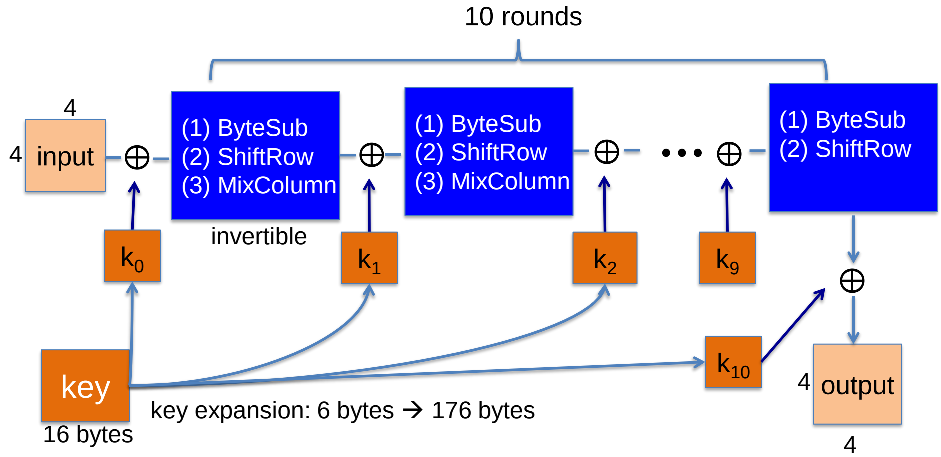
(not a Feistel network)

Recall: AES is prod. cipher organised in rounds. If you travel network right to left you decrypt.



Traverse direction depends on key scheduling
This is not a feistel network

AES-128 schematics



each round, while not linear, is invertible

The round function

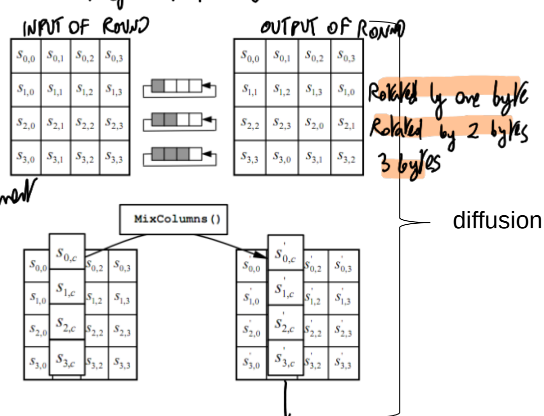
First layer for substitution

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

INPUT OF ROUND   OUTPUT OF ROUND

- **ShiftRows:**
  introduces diffusion
  4x4 matrix with each element a byte

- **MixColumns:**
  (linear transformations)



Rotated by one byte
Rotated by 2 bytes
3 bytes

This column in this slide is obtained through linear transformation of the other columns.

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/

AES security - best known attacks

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

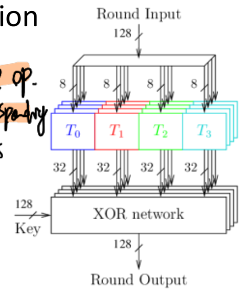
Just 4 times better. Not meaningful

- "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

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

Precompute op. in their corresponding lookup tables



AES good performance also on software implementations

AES Implementation (2/2)

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

Code size/performance tradeoff

Different implementations

	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

17

Example: Javascript AES

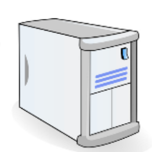
(Stanford Javascript Crypto Library)

AES in the browser:

Download from server

AES library (6.4KB)

no pre-computed tables



Prior to encryption:
pre-compute tables

Then encrypt using tables And precompute on your device (if possible)

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

18

AES in hardware

There are processors that have instructions implementing AES (single round of AES)

- 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 (10 rounds)
    - 9x aesenc + aesenclast
  - Claim 14x speed-up over OpenSSL on the same hw
- Similar instructions for AMD Bulldozer

19

Can be used for building stream cyphers and hash functions