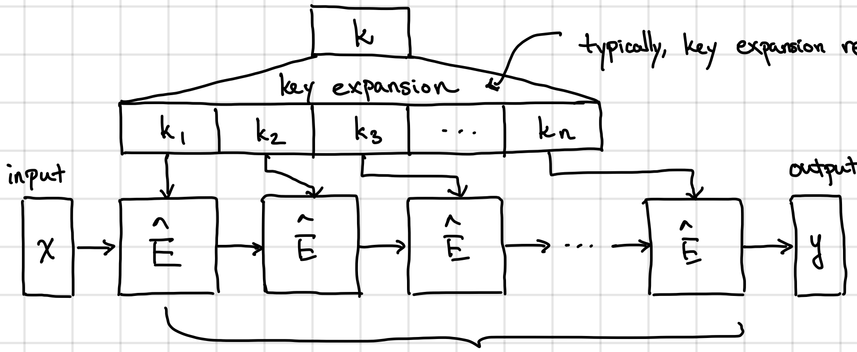
Constructing block ciphers: typically, relies on an "iterated cipher"



typically, key expansion relies on a PRG

$\hat{E}$: round function

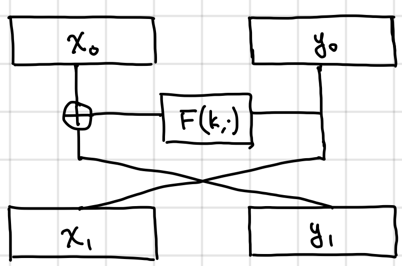
Difficult to design! Never invent your own crypto – use well-studied, standardized constructions and implementations!

We will look at two classic designs:

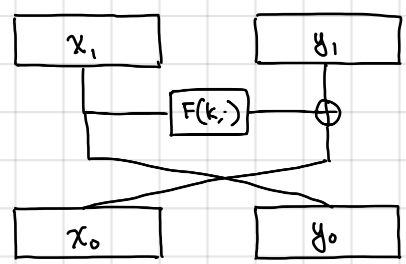
- DES / 3DES (Data Encryption Standard) 1977 (developed at IBM)
- AES (Advanced Encryption Standard) 2002 [most widely used block cipher, implemented in hardware in Intel processors]

on modern Intel processors, (with AES-NI), ~4 cycles/round

DES: relies on the Feistel design:



Evaluation



Inversion

Can stack multiple layers together

Observe: the function F does not have to be invertible $\Rightarrow$ Feistel network is still invertible!

Theorem (Luby-Rackoff). If F is a secure PRF, then a 3-round Feistel construction yields a secure PRP. Similarly, a 4-round Feistel construction implements a strong PRP.

a PRP where the adversary can also query the inversion oracle (i.e., $F^{-1}(k,\cdot)$ in the pseudorandom world and $f^{-1}(\cdot)$ in the random world)

Shows that Feistel construction is sound for constructing block cipher (but now need a good random-looking function F)

called the round function

DES round function will not be a PRF, so overall construction relies on more rounds (but general design philosophy supported by theory)

DES: block size: 64 bits → round function operates on 32-bit blocks

key size: 56 bits (to comply with export control regulations)

used to derive 16 round keys (48 bits)

DES overall is a 16-round Feistel network

simple approach: each 48-bit key is subset of the original 56-bit key

see Boneh-Shoup for description of DES round function

56-bit keys was a compromise between 40-bit keys (NIST/NSA) and 64-bit keys (cryptographers - notably Hellman)

↳ turned out to be insufficient

- 1997: DES challenge solved in 96 days (massive distributed effort)
- 1998: with dedicated hardware, DES can be broken in just 56 hours → not secure enough!
- 2007: using off-the-shelf FPGAs (120), can break DES in just 12.8 days → anyone can now break DES!

↳ 2-DES: apply DES twice (keys now 112-bits)

↳ meet-in-the-middle attack gives no advantage (though space usage is high)

↳ 3-DES: apply DES three times [$3DES((k_1,k_2,k_3),x) := DES(k_3, DES^{-1}(k_2, DES(k_1,x)))$]

↳ 168-bit keys — standardized in 1998 after brute force attacks on DES shown to be feasible

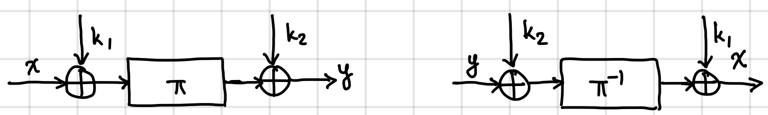
<u>AES</u> (2002 - most common block cipher in use today):

- 3DES is slow (3x slower than DES)
- 64-bit block size not ideal (recall that block size determines adversary's advantage when block cipher used for encryption)

AES block cipher has 128-bit blocks (and 128-bit keys ← also have 192-bit and 256-bit variants (but block size always 2^{128}))

↳ follows another classic design paradigm: iterated Even-Mansour (also called alternating key ciphers)

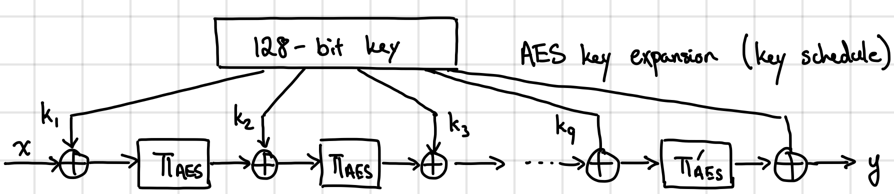
Even-Mansour block cipher: keys (k_1,k_2), input x:



Evaluation   Inversion

<u>Theorem (Even-Mansour)</u>: If π is modeled as a random permutation, then the Even-Mansour block cipher is secure (i.e., it is a secure PRP).

The AES block cipher can be viewed as an iterated Even-Mansour cipher:



key-size ↓

- AES-128: 10 rounds
- AES-192: 12 rounds
- AES-256: 14 rounds

(block-size all 128 bits)

Permutations π_{AES} and π'_{AES} are fixed permutations and <u>cannot</u> be ideal permutations

↳ cannot write down random permutation over $\{0,1\}^{128}$

↳ Cannot appeal to security of Even-Mansour for security

↳ But still provides evidence that this design strategy is <u>viable</u> [similar to DES and Luby-Rackoff]

AES round permutation: composed of three invertible operations that each operate on a 128-bit block

$$\underbrace{\begin{array}{|c|c|c|c|}\hline a_0 & a_1 & a_2 & a_3 \\ \hline a_4 & a_5 & a_6 & a_7 \\ \hline a_8 & a_9 & a_{10} & a_{11} \\ \hline a_{12} & a_{13} & a_{14} & a_{15} \\ \hline \end{array}}$$

128 bits arranged in 4-by-4 grid of bytes ($\{0,1\}^8$)

SubBytes: apply a <u>fixed</u> permutation $S: \{0,1\}^8 \to \{0,1\}^8$ to each cell

↳ hard coded in the AES standard (similar to S-box)

(chosen very carefully to resist attacks)

ShiftRows: cyclic shift the rows of the matrix

- 1st row unchanged
- 2nd row shifted left by 1
- 3rd row shifted left by 2
- 4th row shifted left by 3

MixColumns: the matrix is interpreted as a 4-by-4 matrix over $GF(2^8)$ and multiplied by a fixed <u>invertible</u> matrix (also carefully chosen and hard-coded into the standard)

elements are polynomials over $GF(2)$ ($\mathbb{F}_2$) modulo the irreducible polynomial $x^8 + x^4 + x^3 + x + 1$

<u>Observe</u>: Every operation is invertible, so composition is also invertible

Π_{AES} : SubBytes ; ShiftRows ; MixColumns

Π'_{AES} : SubBytes ; ShiftRows   No MixColumns for the last round [done so AES decryption circuit better resembles AES encryption]

<u>Security of AES</u>: Brute-force attack : 2^{128}

Best-known key recovery attack : $2^{126.1}$ time — only 4x better than brute force!

What does 2^{128}-time look like?

- Suppose we can try 2^{40} keys a second.
  - ↳ 2^{88} seconds to break 1 AES key ~ 10^{19} years (710 million times larger than age of the universe!)
- Total computing power on Earth (circa 2015)
  - ↳ estimated to be ~2^{70} operations/second (currently, bitcoin mining computes ~ 2^{66} hashes/second)

  Let's say we can do 2^{80} operations/second

  ↳ still require 2^{48} seconds to break AES ~ 9 million years of compute

If we move to 256-bit keys, best brute force attack takes $2^{254.2}$ time (on AES-256)

In well-implemented systems, the cryptography is not the weak point — breaking the crypto requires new <u>algorithmic</u> (e.g., quantum computers) techniques

↳ But side channels / bad implementations can compromise crypto