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

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]

DES: relies on the Feistel design:

**Theorem (Luby-ROckett):** 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.

$\rightarrow$ a PRP where the adversary can also query the inversion oracle $F^{-1}(c)$ in the pseudorandom world and $f^{-1}(c)$ in the random world

$\Rightarrow$ 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 $\Rightarrow$ round function operates on 32-bit blocks

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

$\downarrow$ used to derive 16 round keys (48 bits)

$\downarrow$ DES overall is a 16-round Feistel network

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

$\rightarrow$ Can stack multiple layers together
Permutations

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

- 1977: 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 (20), can break DES in just 0.8 days -> anyone can now break DES!

1. 2-DES: apply DES twice (keys now 112-bits)
   - meet-in-the-middle attack gives no advantage (though space usage is high)

2. 3-DES: apply DES three times \[3DES(k_3,k_2,k_1,x) = DES(DES^{-1}(DES(k_3,DES(k_2,DES(k_1,x))))\]
   - 168-bit keys — standardized in 1998 after brute force attacks on DES shown to be feasible

AES (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)
- also have 192-bit and 256-bit variants

AES block cipher has 128-bit blocks (and 128-bit keys)

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

**Even-Mansour black cipher:** keys \((k_1,k_2)\), input \(x\):

```
\[
\begin{array}{c}
X \\
\downarrow \\
\oplus \\
\downarrow \\
\pi \\
\downarrow \\
\oplus \\
\downarrow \\
k_1 \\
\rightarrow y
\end{array}
\]
```

Evaluation

```
\[
\begin{array}{c}
y \\
\downarrow \\
\oplus \\
\downarrow \\
\pi^{-1} \\
\downarrow \\
\oplus \\
\downarrow \\
k_1 \\
\rightarrow X
\end{array}
\]
```

Inversion

**Theorem (Even-Mansour):** If \(\pi\) 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:

```
\[
\begin{array}{c}
\text{128-bit key} \\
\downarrow \\
\text{AES key expansion (key schedule)} \\
\downarrow \\
k_1 \\
\oplus \\
\downarrow \\
\box{T AES} \\
\downarrow \\
k_2 \\
\oplus \\
\downarrow \\
\box{T AES} \\
\downarrow \\
k_3 \\
\oplus \\
\downarrow \\
\box{T AES} \\
\downarrow \\
k_4 \\
\oplus \\
\downarrow \\
\box{T AES} \\
\downarrow \\
k_5 \\
\oplus \\
\downarrow \\
\box{T AES} \\
\downarrow \ldots \\
\box{T AES} \\
\downarrow \\
y
\end{array}
\]
```

Permutations \(T_{AES}\) and \(T'_{AES}\) are fixed permutations and cannot be ideal permutations

1. Cannot appeal to security of Even-Mansour for security
2. But still provides evidence that this design strategy is viable [similar to DES and key-recovery]

\(\rightarrow\) cannot write down random permutation over \(\{0,1\}^{128}\)
AES round permutation: composed of three invertible operations that each operate on a 128-bit block

\[
\begin{array}{cccc}
  a_0 & a_1 & a_2 & a_3 \\
  a_4 & a_5 & a_6 & a_7 \\
  a_8 & a_9 & a_{10} & a_{11} \\
  a_{12} & a_{13} & a_{14} & a_{15}
\end{array}
\]

SubBytes: apply a fixed permutation \( S: \{0,1\}^8 \rightarrow \{0,1\}^8 \) 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 \( \text{GF}(2^4) \) and multiplied by a fixed invertible matrix (also carefully chosen and hard-coded into the standard)

\[
(\begin{array}{cccc}
  2 & 3 & 1 & 1 \\
  1 & 2 & 3 & 1 \\
  1 & 1 & 2 & 3 \\
  3 & 1 & 1 & 2
\end{array})
\]

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

Observe: Every operation is invertible, so composition is also invertible

\text{TIAES} : \text{SubBytes} \; ; \; \text{ShiftRows} \; ; \; \text{MixColumns}

\text{TIAES} : \text{SubBytes} \; ; \; \text{ShiftRows} \; \neg \; \text{MixColumns} \; \text{for the last round.} \; \text{[done so AES decryption circuit better]} \; \text{[resembles AES encryption]}

Security of AES:
- 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.
  \( \Rightarrow 2^{88} \) seconds to break 1 AES key \( \approx 10^9 \) years (710 million times longer than age of the universe!)
- Total computing power on Earth (circa 2015)
  \( \Rightarrow \) estimated to be \( \approx 2^{80} \) operations/second (currently, bitcoin mining computes \( \approx 2^{66} \) hashes/second)
  Let's say we can do \( \approx 2^{80} \) operations/second
  \( \Rightarrow \) still require 2 seconds to break AES \( \approx 9 \) million years of compute

If we move to 256-bit keys, best brute force attack takes \( 2^{1532} \) time (on AES-256)

In well-implemented systems, the cryptography is not the weak point — breaking the crypto requires new algorithmic techniques
- But side channels/bad implementations can compromise crypto