Radboud University

Block Ciphers

Cryptography, Spring 2020

L. Batina, J. Daemen

February 19, 2020

Institute for Computing and Information Sciences
Radboud University

Outline

Block encryption

Block cipher model and security definition

Data Encryption Standard (DES)

Rijndael and AES

Block encryption

The trouble with stream encryption

Stream encryption requires attention

(1) Diversifier collisions are fatal and avoiding them is seen as difficult
  - taking a counter for D:
    - implies *keeping state* in between messages
    - in some architectures this is problematic
  - generating D randomly
    - generating high quality randomness is hard
    - there remains a risk of collisions

(2) It does not protect integrity of the plaintext
  - adversary can flip individual bits in ciphertext
  - …flipping corresponding bits in plaintext
  - this is likely to go undetected by message recipient

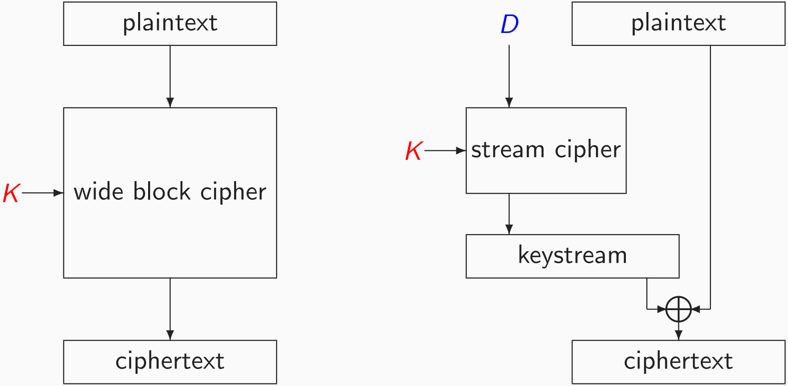
Some see the answer to these issues in different type of encryption

Block Encryption

Block encryption, ideally

- Encryption as a scrambling recipe
  - transforming the full plaintext by a sequence of operations
  - (some of) these transformations depend on a secret key K
  - it must be invertible: there must be a recipe for decryption
  - ciphertext is as long as the plaintext (. . . or a little longer)
- Such a recipe is called a wide block cipher, considered secure if:
  - it maps similar plaintexts to seemingly unrelated ciphertexts
  - . . . and vice versa
  - and this map is completely different for different keys K
- How does this address concerns of stream encryption?
  - *similar* plaintexts give unrelated ciphertexts, so *no need for* D
  - small changes in ciphertext give a completely different plaintext
- But . . .
  - some leakage remains: equal messages give equal ciphertexts
  - tamper detection isn't absolute: requires redundant plaintext
  - what about protection against replay attacks?

Block vs. stream encryption illustrated

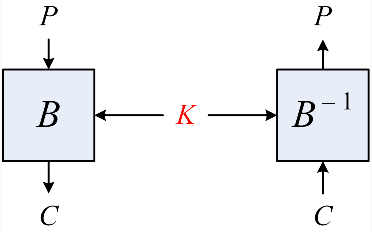


Block encryption in practice

- Problem: building a wide block cipher may be hard
  - until a few years ago in experimental stage
  - as *modes* of underlying primitives
- The established block ciphers have fixed length
  - best known: DES (8-byte plaintexts) and AES (16-byte)
  - longer plaintexts require splitting in blocks and padding
  - ...and the application of modes (see later)
- By fixing length, *advantages* of block encryption evaporate
  - unless we tolerate more leakage
  - redundancy in plaintext no longer provides integrity
- But we still will treat block ciphers in this course
  - most real world symmetric crypto still based on block ciphers
  - we'll look upon them as we do now on rotor machines

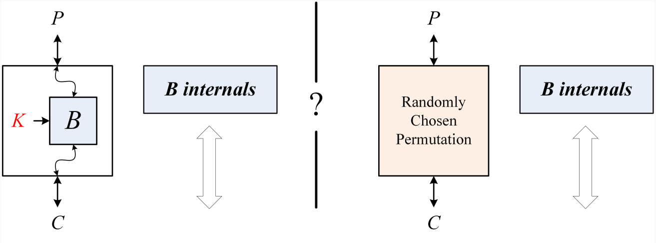
Block cipher model and security definition

Block cipher definition



- Permutation B operating on $\{0,1\}^b$ with b the block length
  - parameterized by a secret key: B_K
  - with an inverse B_K^{-1} that should be efficient
- Computing $C = \mathrm{B}_K(P)$ or $P = \mathrm{B}_K^{-1}(C)$ should be
  - efficient knowing the secret key K
  - infeasible otherwise
- Dimensions: block length b and key length $|K|$

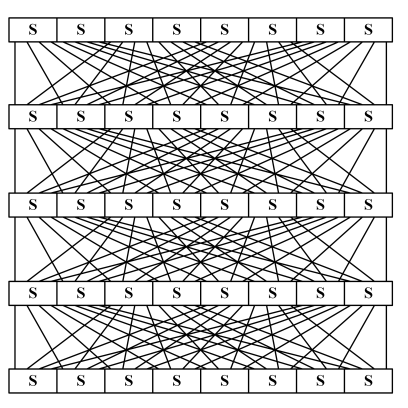
Pseudorandom Permutation (PRP) security



- Infeasibility to distinguish B_K from randomly chosen permutation
- Advantage as $\epsilon(M, N)$
  - Q_s to $\mathrm{B}_K^{(-1)}$ or $\mathrm{RCP}^{(-1)}$: online or data complexity M
  - Q_c to B internals: offline or computational complexity N
- PRP if adversary can only make forward queries, SPRP if both forward and inverse ones
- Per default, block cipher considered secure if SPRP $\mathrm{Adv} = N2^{-|K|}$

Data Encryption Standard (DES)

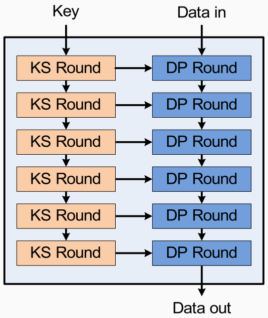
Product cipher [Claude Shannon, 1949] and SPN



Round function in data path with two (or three) layers

- non-linear substitution layer: S-boxes applied in parallel
- permutation layer: moves bits to different S-box positions
- K?: either key-dependent S-boxes or third layer of *key addition*

Iterative block ciphers

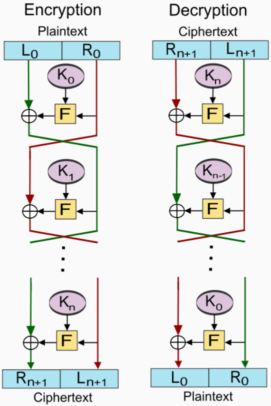


- Data path (right): transforms input data to output data
  - iteration of a non-linear round function
  - . . . that depends on a round key
- Key schedule (left)
  - generates round keys from cipher key K

Data encryption standard (DES)

- Standard by and for US government
- By National Institute for Standardization and Technology (NIST)
- Designed by IBM in collaboration with NSA
- 1977: Federal Information Processing Standard (FIPS) 46
  - complete block cipher specification
  - block length: 64 bits, key length: 56 bits
  - no design rationale
  - freely usable
- Massively adopted by banks and industry worldwide
- Dominated symmetric crypto for more than 20 years

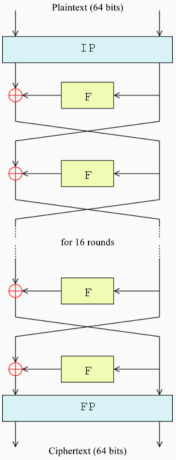
The Feistel structure



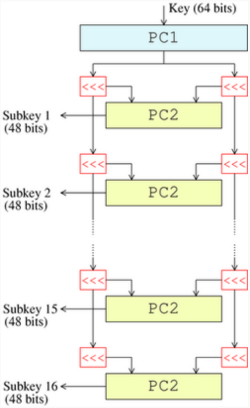
- State: left half L and right half R
- Alternation of involutions
  - apply F to R_i and add to L_i
  - swap left and right
- Omit swap in last round
- B^{-1} similar to B
  - same operation sequence
  - round keys in reversed order
- No need for F^{-1}
- Used in DES

Data encryption standard: overview

data path

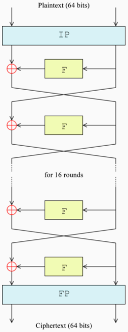


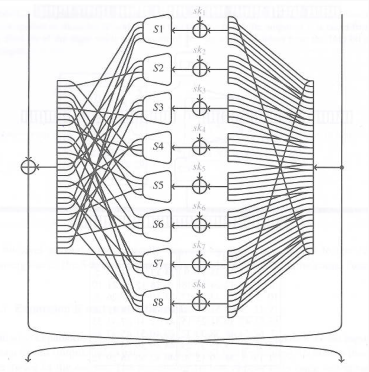
key schedule



DES algorithmic structure

- Data path
  - 16-round Feistel
  - Initial (IP) and final permutations (FP):
    - no cryptographic significance
    - historical, due to addressing in hardware implementation
- Key schedule
  - 8 bits thrown away in permuted choice 1 (PC1)
  - remaining 56-bit string
    - split in two 28-bit strings
    - rotated for each round over 1 or 2 bits
  - 48-bit round key obtained with PC2 of these 56 bits
  - each round key bit is just a cipher key bit





- Variant of SPN with 4 layers:
  - expansion E: from 32 to 48 bits
  - bitwise round key addition
  - substitution: 8 different 6-to-4 bit non-linear S-boxes
  - permutation P: moving nearby bits to remote positions
- Clearly hardware-oriented

Non-ideal DES property: Weak Keys

- What happens in the case of $K = 0^*$: the all-zero cipher key?
  - all round keys are all-zero
  - all rounds are the same
  - cipher and its inverse are the same
- Same is true for $K = 1^*$: the all-one cipher key
- And two more keys due to symmetry in key schedule
- These keys, including 0^* and 1^* are called *weak keys* K_w:

$$\mathsf{DES}_{K_w} \circ \mathsf{DES}_{K_w} = \mathrm{I}$$

- Also 6 semi-weak key pairs (K_1, K_2)

$$\mathsf{DES}_{K_1} \circ \mathsf{DES}_{K_2} = \mathrm{I}$$

- Mostly of academic interest

Non-ideality in DES: Complementation Property

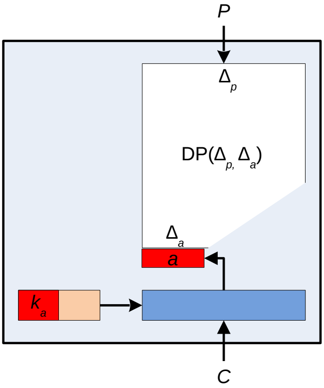
- What happens if we complement the cipher input?
  - flip all bits in key
  - flip all bits in plaintext
- In first round
  - input to F complemented so output of E complemented
  - round key also complemented so input to S-boxes unaffected
  - output of F unaffected
- Output of first round is simply complemented
- Repeat this until you reach the ciphertext
- Complementation property:

$$\mathrm{DES}_K(P) = C \Leftrightarrow \mathrm{DES}_{\overline{K}}(\overline{P}) = \overline{C}$$

- Reduces security strength from 56 to 55 due to speed up of exhaustive key search

Differential cryptanalysis [basic idea, more about it later]

- Statistical attack with following distinguisher:
  - inputs P_i and P_i^* with $P_i \oplus P_i^* = \Delta_p$
  - lead to difference Δ_a at input of last round
  - with relatively high probability $\mathrm{DP}(\Delta_p, \Delta_a)$
- Requires about $1/\mathrm{DP}(\Delta_p, \Delta_a)$ input/output pairs
- Many variants exist



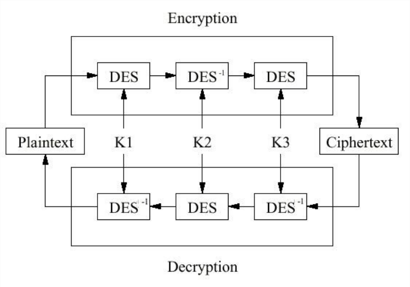
Breaking DES: differential and linear cryptanalysis (DC & LC)

- Differential cryptanalysis attack by Eli Biham and Adi Shamir, 1990
  - Requires $M \approx 2^{47}$ (1000 TeraByte) chosen plaintexts
  - Breaks DES because total complexity for the break is below 2^{55}
  - No real-world relevance: unrealistic amount of data required
- Linear cryptanalysis attack by Mitsuru Matsui, 1992
  - Also statistical attack, *the dual of* DC
  - Requires about $M \approx 2^{43}$ (64 TeraByte) known plaintexts
  - Less data than DC but still unrealistic amount for real-world attack
- Academic relevance:
  - provided first systematic design criterion for block ciphers
  - LC/DC resistance is basis of modern symmetric crypto design

The real problem of DES: the short key

- Exhaustive key search: about 3.6×10^{14} trials
- More than 21 years ago: “software” cracking
  - about 10.000 workstations, each 500.000 trials/second
  - expected time: 7.200.000 seconds: 2,5 months
  - applied in cracking RSA labs DES challenge, June ’97
- Cracking using dedicated hardware
  - COPACOBANA RIVYERA (2008)
  - board with 128 Spartan-3 5000 FPGAs, costs about 10.000$
  - finds a DES key in less than a day
- Following Moore’s law, same budget would now give $<$ 10 minutes
- Short DES key is real-world concern!

Triple DES (FIPS 46-2 and 46-3)



- Double DES allows meet-in-the-middle attacks
- Three variants of Triple-DES
  - 3-key: 168-bit key, only option allowed by NIST
  - 2-key: 112-bit key by taking $K_3 = K_1$
    - still massively deployed by banks worldwide
  - 1-key: 56-bit key by taking $K_3 = K_2 = K_1$

Rijndael and AES

The AES competition

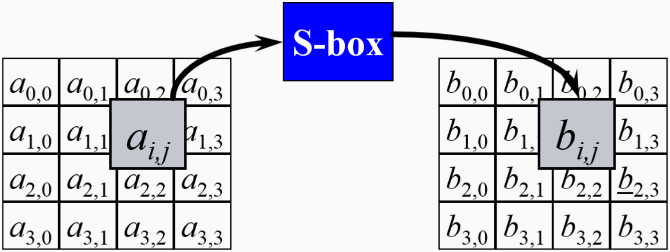
- NIST launches the AES open contest to replace DES in 1997
  - 128-bit block length, 128-, 192- and 256-bit keys
  - specs, code, design rationale and preliminary analysis
  - Vincent Rijmen and I submitted RIJNDAEL
- First round: August 1998 to August 1999
  - 15 candidates at 1st AES conference in Ventura, California
  - analysis presented at 2nd AES conf. in Rome, March 1999
  - NIST narrowed down to 5 finalists using this analysis
- Second round: August 1999 to summer 2000
  - analysis presented at 3rd AES conf. in New York, April 2000
  - NIST selected winner using this analysis: RIJNDAEL
- NIST motivated their choice in two reports

Rijndael

- Block cipher with block and key lengths $\in \{128, 160, 192, 224, 256\}$
  - set of 25 block ciphers
  - AES limits block length to 128 and key length to multiples of 64

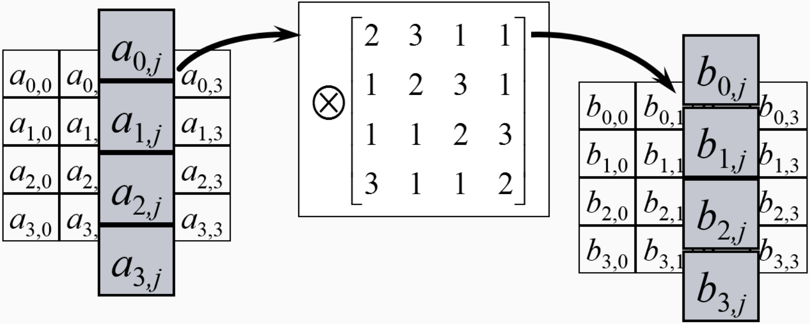
  we only treat AES in this course
- Iteration of a round function with following properties:
  - 4 layers: nonlinear, shuffling, mixing and round key addition
  - all rounds are identical
  - ...except for the round keys
  - ...and omission of mixing layer in last round
  - parallel and symmetric
- Key schedule
  - Expansion of cipher key to round key sequence
  - Recursive procedure that can be done in-place
- Manipulates bytes rather than bits

The non-linear layer: SubBytes



- The same invertible S-box applied to all bytes of the state
- Assembled from building blocks that were proposed and analyzed in cryptographic literature
- Criteria:
  - to offer resistance against DC, LC and *algebraic* attacks
  - ... when combined with the other layers

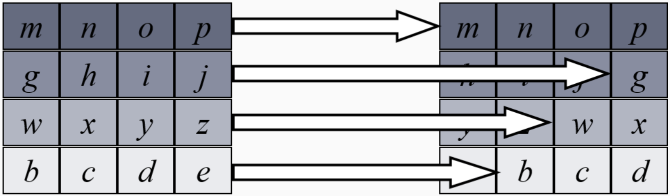
The mixing layer: MixColumns



- Same invertible mapping applied to all 4 columns
- Multiplication by a 4×4 circulant matrix [for info: in $\mathbb{F}_{2^8}$]
  - difference in single input byte propagates to 4 output bytes
  - difference in 2 input bytes propagates to 3 output bytes
  - difference in 3 input bytes propagates to 2 output bytes

$\Rightarrow$ we say: it has *branch number* 5

The shuffling layer: ShiftRows



- Each row is shifted by a different amount
- Different shift offsets for higher block lengths
- Moves bytes in a given column to 4 different columns
- Combined with MixColumns and SubBytes this gives fast diffusion

Round key addition: AddRoundKey

$$\begin{array}{|c|c|c|c|}\hline a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ \hline a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ \hline a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ \hline a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \\ \hline \end{array} + \begin{array}{|c|c|c|c|}\hline k_{0,0} & k_{0,1} & k_{0,2} & k_{0,3} \\ \hline k_{1,0} & k_{1,1} & k_{1,2} & k_{1,3} \\ \hline k_{2,0} & k_{2,1} & k_{2,2} & k_{2,3} \\ \hline k_{3,0} & k_{3,1} & k_{3,2} & k_{3,3} \\ \hline \end{array} = \begin{array}{|c|c|c|c|}\hline b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ \hline b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\ \hline b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ \hline b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \\ \hline \end{array}$$

Round key is computed from the cipher key K

Key schedule: example with 192-bit key K

k_0	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9	k_{10}	k_{11}	k_{12}	k_{13}	k_{14}	k_{15}	$\cdots$

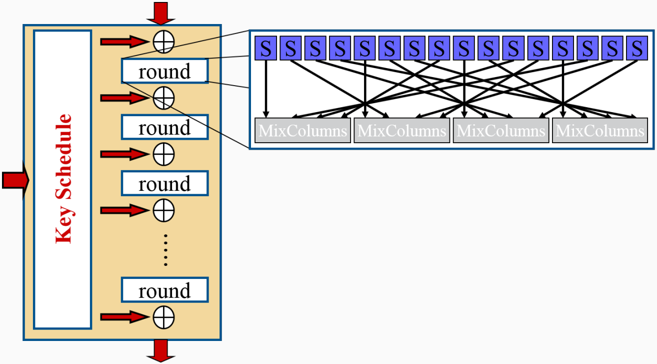
Round key 0	Round key 1	Round key 2	$\cdots$

- Expansion: put K in 1st columns and compute others recursively:

$$\begin{aligned} k_{6n} &= k_{6n-6} \oplus f(k_{6n-1}) \\ k_i &= k_{i-6} \oplus k_{i-1}, \quad i \neq 6n \end{aligned}$$

  with f: 4 parallel AES S-boxes followed by 1-byte cyclic shift
- Selection: round key i is columns $4i$ to $4i+3$

AES: summary



- 10 rounds for 128-bit key, 12 for 192-bit key and 14 for 256-bit key
- Last round has no MixColumns so that inverse is similar to cipher

AES security status anno 2020

- Cryptanalysis with respect to SPRP (in public domain)
  - no attacks of full-round version after 2 decades of intense public scrutiny
  - attacks on reduced-round versions with more than 5 rounds have **huge** data complexity
  - this leads to high assurance about SPRP security of AES
- Implementation attacks: exploiting physical features
  - timing attacks: cache misses in table-lookups
  - power analysis: exploiting dependence of current on data
  - electromagnetic analysis: same for EM emanations
  - fault attacks: exploiting forced faults
- Implementation attacks are the ones that matter in practice!

Summary

Summary

- Block ciphers are keyed b-bit permutations
  - a different permutation B_K per key K
  - with an efficient inverse B_K^{-1}
  - (S)PRP-secure if B_K is hard to distinguish from random permutation
  - exhaustive keysearch should be best method and has success probability $N2^{-|K|}$
- DES and AES are the most widespread block ciphers
  - constructed by iterating a simple round function
  - round has layers for non-linearity, mixing, shuffling and key addition