Crypto 4

Note: this material is not intended to replace the live lecture for students.

Contents

4.1 Block Ciphers

Figure 4.1.1: a Block Cipher

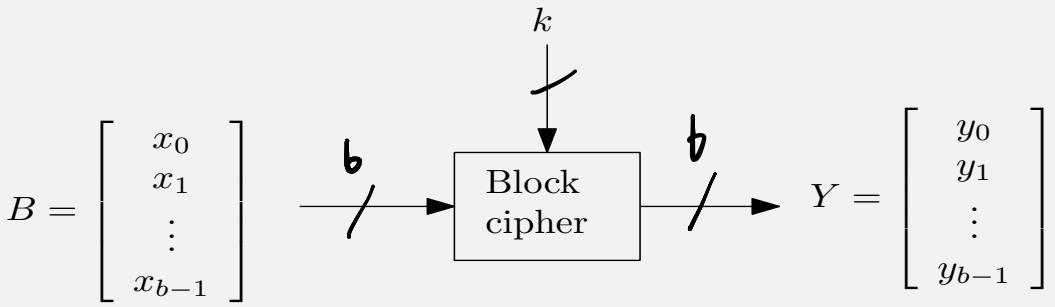
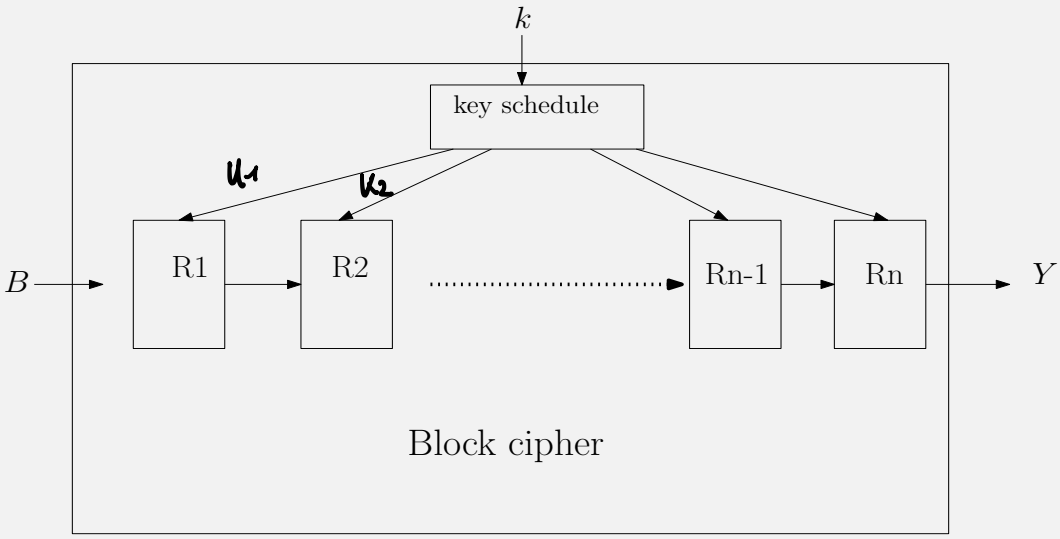


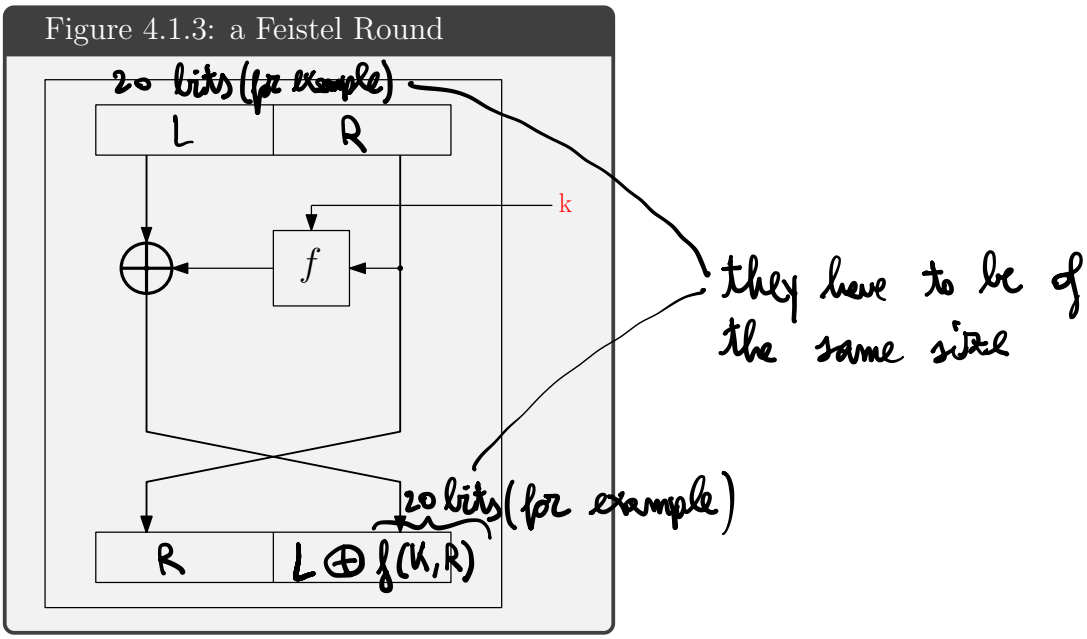
Figure 4.1.2: Rounds



To decipher k16, k15, k14, ...

4.1.1 Feistel

Figure 4.1.3: a Feistel Round

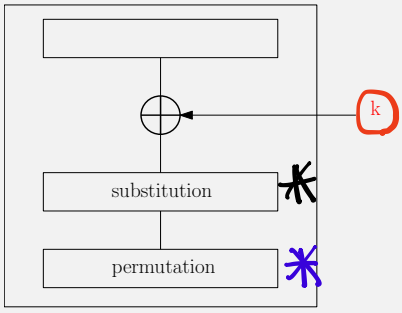


NOTE 4.1.4

Notice that to decipher the Feistel rounds are the same. It is only necessary to invert the order of the subkeys delivered by the Key Schedule algorithm.

4.1.2 SPN (Substitution-Permutation Network)

Figure 4.1.5: a SP Round



The following is a toy block cipher used by Howard M. Heys in his lectures A Tutorial on Linear and Differential Cryptanalysis:

4.1.6 4×4 S-Box

In our cipher, we break the 16-bit data block into four 4-bit sub-blocks. Each sub-block forms an input to a 4×4 S-box (a substitution with 4 input and 4 output bits), which can be easily implemented with a table lookup of sixteen 4-bit values, indexed by the integer represented by the 4 input bits. The most fundamental property of an S-box is that it is a nonlinear mapping, i.e., the output bits cannot be represented as a linear operation on the input bits.

Figure 4.1.7: Toy SPN

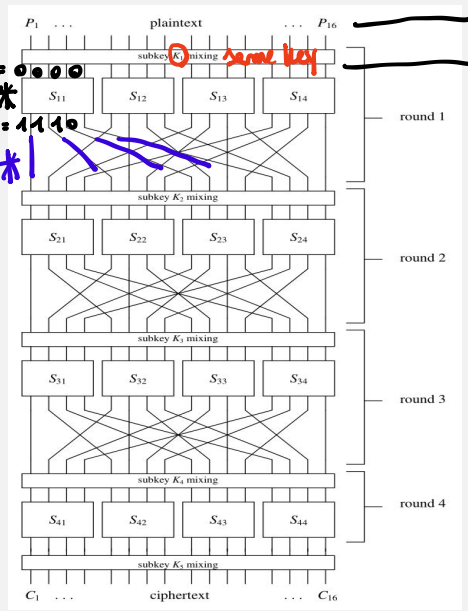


Figure 4.1.8: S-Box (Substitution)

input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
output	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

Figure 4.1.9: Permutation

input	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
output	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16

4.2 DES

The increase in computer communications and the advent of electronic fund-transfer systems in the early 1970s was one of the main factors behind the decision of the United States government to try to impose standards on the security and custody of Federal and other data banks. Accordingly, in 1973–4 the National Bureau of Standards (NBS) advertised for anyone interested to submit proposals for a data encryption standard. The hope/idea was that the encryption process might be fitted to a small chip, and that this would result in a mass-produced, widely used chip whose security was beyond question.

The scheme adopted was that proposed by IBM. It was based on their Lucifer scheme, and was of the same type as described in the last section. However, whereas the Lucifer scheme used a key of 128 bits, the key size in the scheme proposed to NBS was 64 bits, and eight of these bits were thrown away by the encryption algorithm. Based on this proposal the Data Encryption Standard (DES) was published in 1977 by the NBS for use by the Federal government to protect valuable and sensitive but unclassified data.

→ predecessor of NIST

[Welsh89, pag. 165]

AES is byte-oriented, DES is bits-oriented

Figure 4.2.1: DES

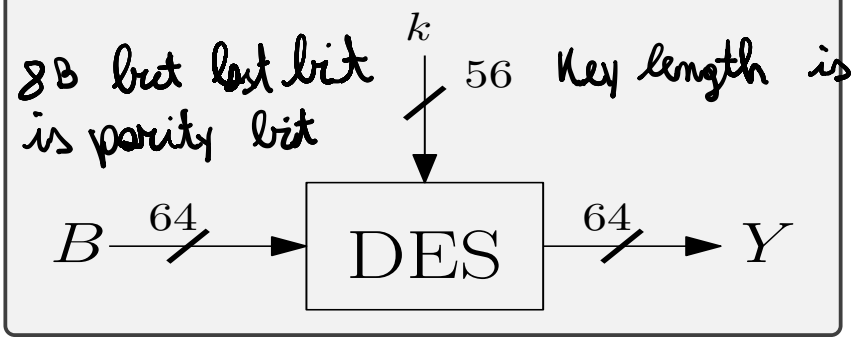
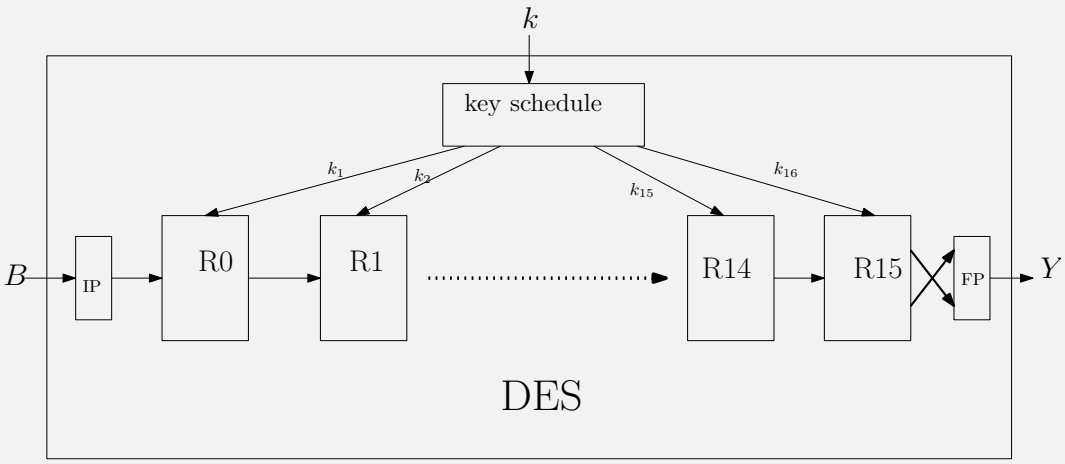


Figure 4.2.2: DES Rounds



NOTE 4.2.3

Besides the 16 rounds DES has an initial permutation IP and a block exchange between the last round and the final permutation $\text{FP} = \text{IP}^{-1}$.

DES rounds are Feistel.

Figure 4.2.4: DES Feistel Round

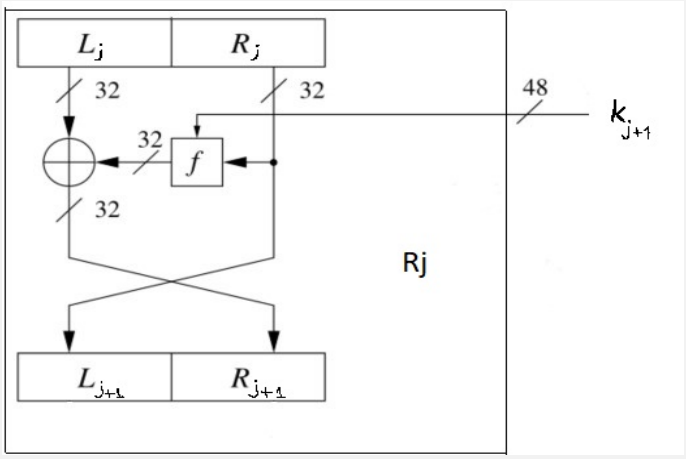
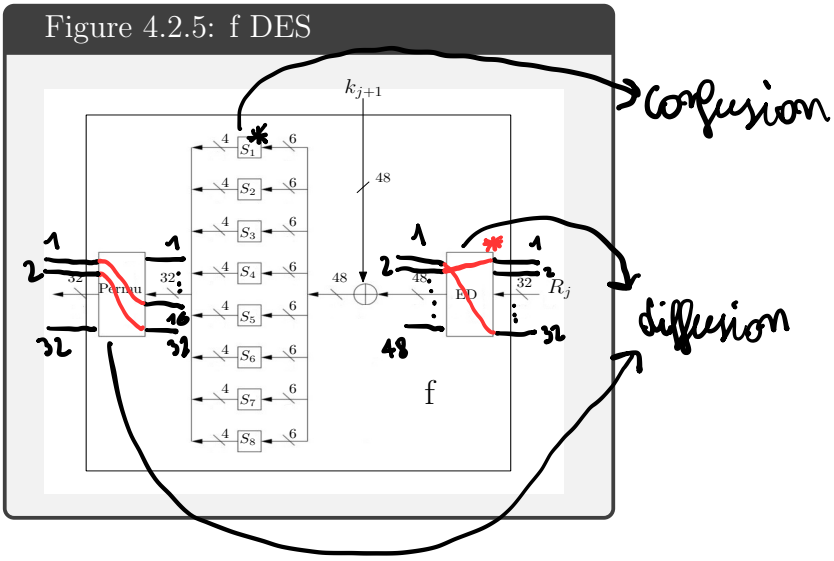


Figure 4.2.5: f DES



DOUBLE ENCRYPTION: $|k| = 56$ bits, $|\tilde{k}| = 56$ bits

$$Enc_{k}\left(Enc_{\tilde{k}}(P)\right) = T_{k\tilde{k}}(P)$$

$k\tilde{k}$: 56 + 56 = 112

P_0, C_0 such that $T_{k\tilde{k}}(P_0) = C_0$ ← known to the adversary

↳ recover the keys k and $\tilde{k}$ ←

the attack called meet in the middle → 57 bits ← 2^{57} iterations (worst case)

$T_{k\tilde{k}}(P_0) = Enc_k(Enc_{\tilde{k}}(P_0)) = C_0$

$k\tilde{k}$ = k's

k_1	$Enc_{k_1}(P_0)$
k_2	$Enc_{k_2}(P_0)$
k_3	$Enc_{k_3}(P_0)$
⋮	
s	$Enc_s(P_0)$
⋮	
$k_{2^{56}}$	$Enc_{k_{2^{56}}}(P_0)$

(P_0, C_0) known to adversary
= KPA (Known Plaintext Attack)

$Dec_{k'}(C_0)$ → run 2^{56} iter. in worst case to decrypt

2^{56} iterations to construct the table

Meet-in-the-middle $\Rightarrow Enc_s(P_0) = Dec_{k'}(C_0)$

$$Enc_{k'}(Enc_s(P_0)) = C_0$$

$$2^{56} + 2^{56} = 2 \cdot 2^{56} = 2^{57}$$

2^{56} → to construct the table

2^{56} → to decrypt

NOTE 4.2.6

The eight S-box (substitution box) $S_1, \cdots, S_8$ are different from each other.

S-boxes are lookup tables i.e. a map from input to output which uses the input as an index to lookup.

Here some of the guiding ideas about the design of the S-box:

1. No single output bit should be too close to a linear combination of the input bits.
2. If the lowest and the highest bits of the input are fixed and the four middle bits are varied, each of the possible 4-bit output values must occur exactly once.
3. If two inputs to an S-box differ in exactly one bit, their outputs must differ in at least two bits.
4. If two inputs to an S-box differ in the two middle bits, their outputs must differ in at least two bits.
5. If two inputs to an S-box differ in their first two bits and are identical in their last two bits, the two outputs must be different.
6. For any nonzero 6-bit difference between inputs, no more than 8 of the 32 pairs of inputs exhibiting that difference may result in the same output difference.
7. A collision (zero output difference) at the 32-bit output of the eight S-boxes is only possible for three adjacent S-boxes.

Here is S_1: 4 bits output

S_1	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0yyyy1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
1yyyy0	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
1yyyy1	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

6 bits input 110101

4 bits output 0011 (3_{10})

boom

For other DES S-Box see https://en.wikipedia.org/wiki/DES_supplementary_material.

Exercise 4.2.7

Check properties 2.,3.,4.,5. for S-box S_1.

The S-boxes are the most crucial elements of DES because they introduce a *non-linearity* to the cipher, i.e.,

$$S(a) \oplus S(b) \neq S(a \oplus b).$$

Without a nonlinear building block, an attacker could express the DES input and output with a system of linear equations where the key bits are the unknowns.

[Paar10, pag. 66]

Exercise 4.2.8

Check $S_1(4) \oplus S_1(23) = S_1(4 \oplus 23)$.

Exercise 4.2.9

Check $S_1(0) \neq 0$. This shows that S_1 is non linear.

NOTE 4.2.10

The permutation Perm of f introduce "Shannon's difussion". Namely, the output 4 bits of the S-box are permuted so they hit several S-boxes in the next round.
Thus, the combined effect of the expansor ED, the S-boxes and the permutation Perm assure that every bit of the output block y has a strong dependence on all bits of the key and all bits of the input block x. This is the so called avalanche effect.

The 32nd in input with the 1st output

ED

1	2	3	4	5	6
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

numbers are repeated because input bits are less than output bits

Permu

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Exercise 4.2.11

Compare the output of f for inputs $R_j = 0$ and $R_j = 1$ with $k_{j+1} = 0$. You should use the S-boxes in https://it.wikipedia.org/wiki/Informazioni_aggiuntive_su_DES.

4.2.1 DES is not secure!

> It is fairly safe to assert that no other cryptographic device has aroused such a controversy as the Data Encryption Standard. M. E. Hellman and W. Diffie of Stanford University were the first to observe that the key length of 56 effective bits was uncomfortably short and suggested an increase to 64 bits, or even 128 bits as in the Lucifer scheme. As a result of this and an article by D. Kahn (New York Times, 3 April 1976), the NBS held two workshops to 'answer the criticisms'.

[Welsh89, page 167]

This means that a KPA brute-force is computationally feasible.

Neither 2DES is secure due to the attack: meet-in-the-middle !!

3DES is considered "secure". 3DES is usually implemented as EDE:

$$\mathbf{DES}_{k_3} \circ \mathbf{DES}^{-1}_{k_2} \circ \mathbf{DES}_{k_1} \quad \sim 128 \text{ bits security}$$

NOTE 4.2.12

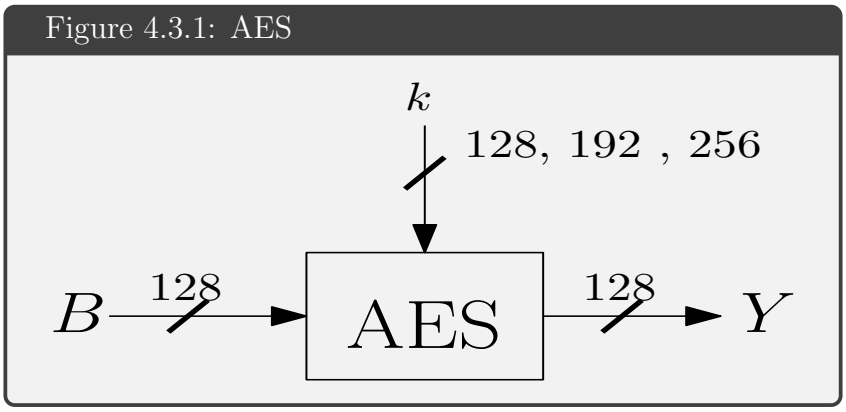
Here is a nice video where Whitfield Diffie, Martin Hellman, Ron Rivest, Adi Shamir and Dickie George talk about DES: https://www.youtube.com/watch?v=0NlZpyk3PKI

4.3 AES

AES is byte oriented. Blocks of $128 = 2^4 \cdot 2^3$ bits i.e. 16 bytes. better to think in Bytes
The algorithm, also known as Rijndael, was designed by the belgian cryptographers Joan Daemen e Vincent Rijmen.

AES is an specific implementation of Rijndael, adopted by the National Institute of Standards and Technology (NIST) in november 2001 after 5 years of study, proves and a final selection between several algorithms.

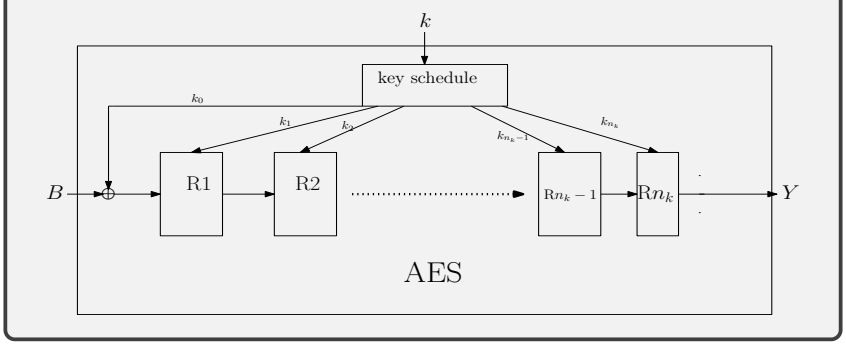
Figure 4.3.1: AES



The number of AES rounds depends upon the key:

k	♯ Rounds
128 bits	10
192 bits	12
256 bits	14

Figure 4.3.2: AES Rounds

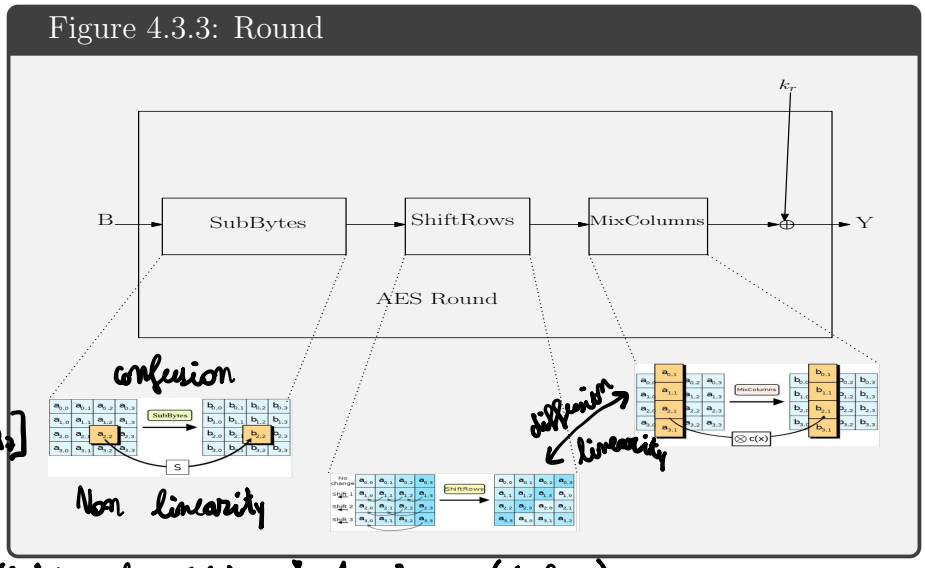


Each block $B = [b_0 \cdots b_{15}]$ is handle 4×4 matrix of 16 bytes:

$$B = \begin{bmatrix} b_0 & b_4 & b_8 & b_{12} \\ b_1 & b_5 & b_9 & b_{13} \\ b_2 & b_6 & b_{10} & b_{14} \\ b_3 & b_7 & b_{11} & b_{15} \end{bmatrix}$$

$16 \cdot 8 = 128$ bits

Figure 4.3.3: Round



$a_{2,2}$ is a byte

$S(a_{2,2}) = b_{2,2}$

$a_{2,2}$ byte of 7^{th}

$a_{2,2} = [a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7]$

$a_0 + a_1 x + a_2 x^2 + \ldots + a_7 x^7$

reminder of modular division by $G(x) = x^8 + x^4 + x^3 + x + 1$ (Galois)

highly non* linear op.

The bytes $b_0, \cdots, b_{15}$ are actually regarded as elements of the finite Galois field $\mathrm{GF}(2^8) = \mathbb{F}_{2^8}$ constructed as the remainders of the modular polynomial $P(x) = x^8 + x^4 + x^3 + x + 1$. Usually the hexadecimal notation is used e.g. $b = \mathrm{A3} \in \mathbb{F}_{2^8}$ stands for the polynomial $x^7 + x^5 + x + 1$.

$\Rightarrow (r_0 + r_1 x + \ldots + r_7 x^7)(a_0 + a_1 x + \ldots + a_7 x^7) \equiv 1 \mod (G(x) = x^8 + x^4 + x^3 + x + 1)$

$S(a_{2,2}) \rightarrow r_0 + r_1 x + \ldots + r_7 x^7$

$= [s_0 s_1 \ldots s_7]$

$A \cdot \begin{bmatrix} r_0 \\ r_1 \\ \vdots \\ r_7 \end{bmatrix} + v = \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_7 \end{bmatrix}$

The mixColumns is also related to a modular polynomial : $Q(z) = z^4 + 1$. Each column $\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$ is regarded as a polynomial $a_0 + a_1 z + a_2 z^2 + a_3 z^3$ and the mixColumn $\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \rightarrow \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$ is as follows:

$$b_0 + b_1 z + b_2 z^2 + b_3 z^3 := (3z^3 + z^2 + z + 2) \times (a_0 + a_1 z + a_2 z^2 + a_3 z^3) \pmod{z^4 + 1}$$

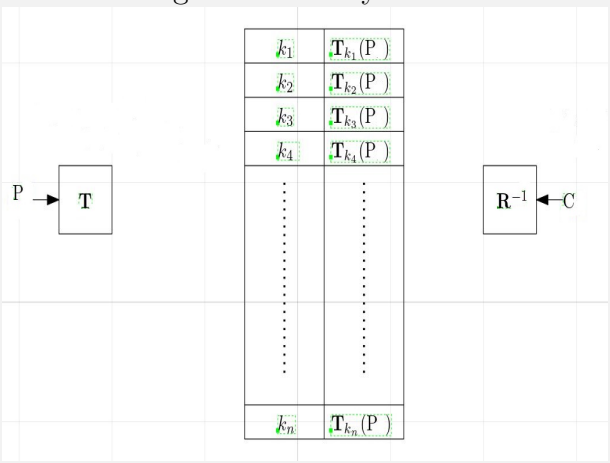
4.4 Attacks

4.4.1 Double encryption & Meet-in-the-Middle

Meet-in-the-middle

Double encryption means that a block B is cipher twice by using two block ciphers (and two keys) R, T. So the key dimension is the sum of the two key-bits dimensions. This can lead to the wrong idea that the security level is substantially improve by a double encryption.

This is wrong as showed by the **meet-in-the-middle** attack:



Exercise 4.4.1

Let $|k_T|, |k_R|$ be the key length of the ciphers T, R respectively. Show that $2^{|k_R|} + 2^{|k_T|}$ iterations are necessary to perform meet-in-the-middle.

Exercise 4.4.2

Why meet-in-the-middle does not work for triple encryption ? Hint: count how many iterations are necessary to perform the attack.

4.5 Bibliography

Books I used to prepare this note:

[Aumasson18] Jean-Philippe Aumasson, *Serious Cryptography: A Practical Introduction to Modern Encryption*, No Starch Press, 2018.

[KatLin15] Jonathan Katz; Yehuda Lindell, *Introduction to Modern Cryptography* Second Edition, Chapman & Hall/CRC, Taylor & Francis Group, 2015.

[Paar10] Paar, Christof, Pelzl, Jan, *Understanding Cryptography, A Textbook for Students and Practitioners*, Springer-Verlag, 2010.

[Welsh89] Dominic Welsh, *Codes and Cryptography* Oxford Universitary Press, New York, 1989.

Here a list of papers:

[DH76] Diffie, W.; Hellman, M. *New directions in cryptography*, (1976). IEEE Transactions on Information Theory. 22 (6): 644654.

[Shannon49] C.E. Shannon, *Communication Theory of Secrecy Systems*, Bell System Technical Journal, vol. 28 -4. October 1949 pp 656-715.

and some interesting links:

https://en.wikipedia.org/wiki/Advanced_Encryption_Standard