

Συμμετρικά Κρυπτοσυστήματα

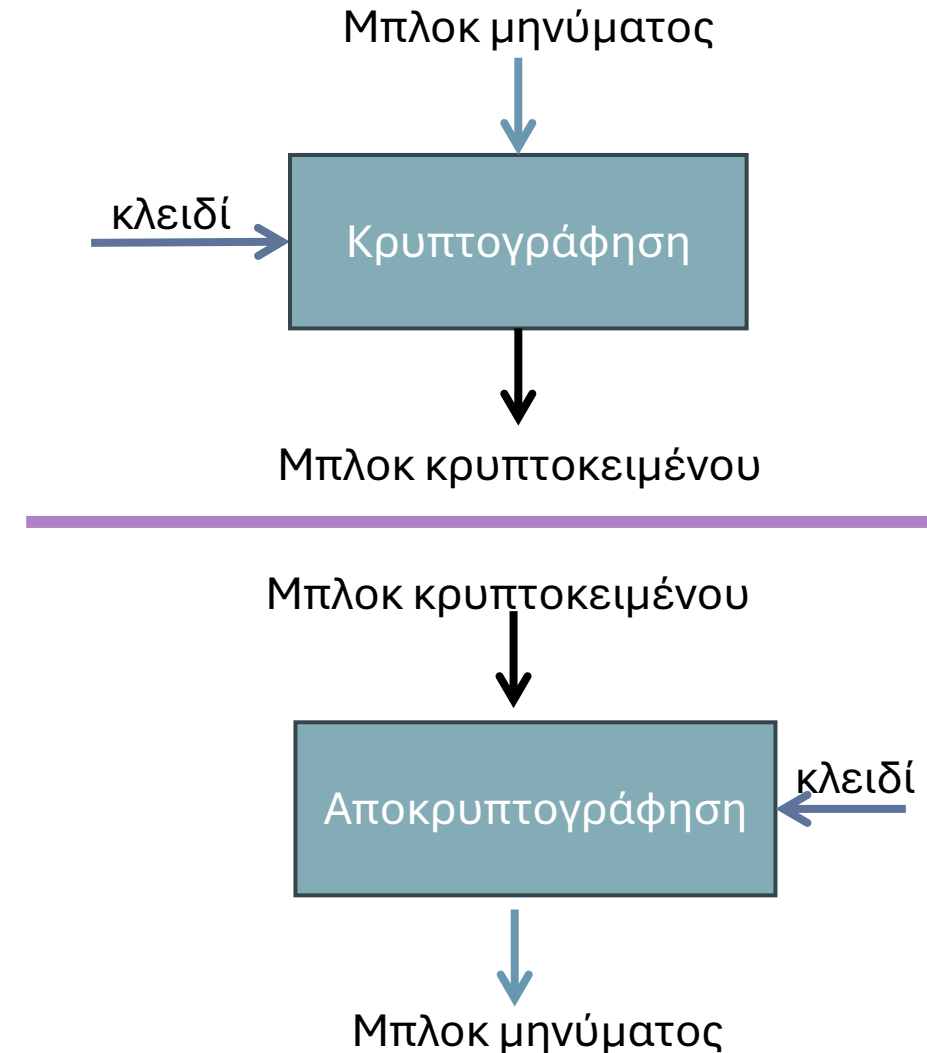
Παναγιώτης Γροντάς
ΕΜΠ Κρυπτογραφία
2026 – 2027

Με βάση παλιότερες διαφάνειες Άρη Παγουρτζή – Στάθη Ζάχου – Πέτρου Ποτίκα



Block Ciphers (Κρυπτοσυστήματα Τμήματος)

- Συμμετρικά κρυπτοσυστήματα
 - Ίδιο κλειδί κρυπτογράφησης
 - Ίδιο κλειδί αποκρυπτογράφησης
- Το μήνυμα χωρίζεται σε τμήματα *ίδιου μεγέθους (block)*
 - Σήμερα *128 bits* (παλιότερα *64 bits*)
- Συνήθως Μέγεθος block $\neq$ Μέγεθος κλειδιού
 - Δεν είναι υποχρεωτικό να συμπίπτουν, αλλά δεν απαγορεύεται



Block Ciphers (Κρυπτοσυστήματα τμήματος)

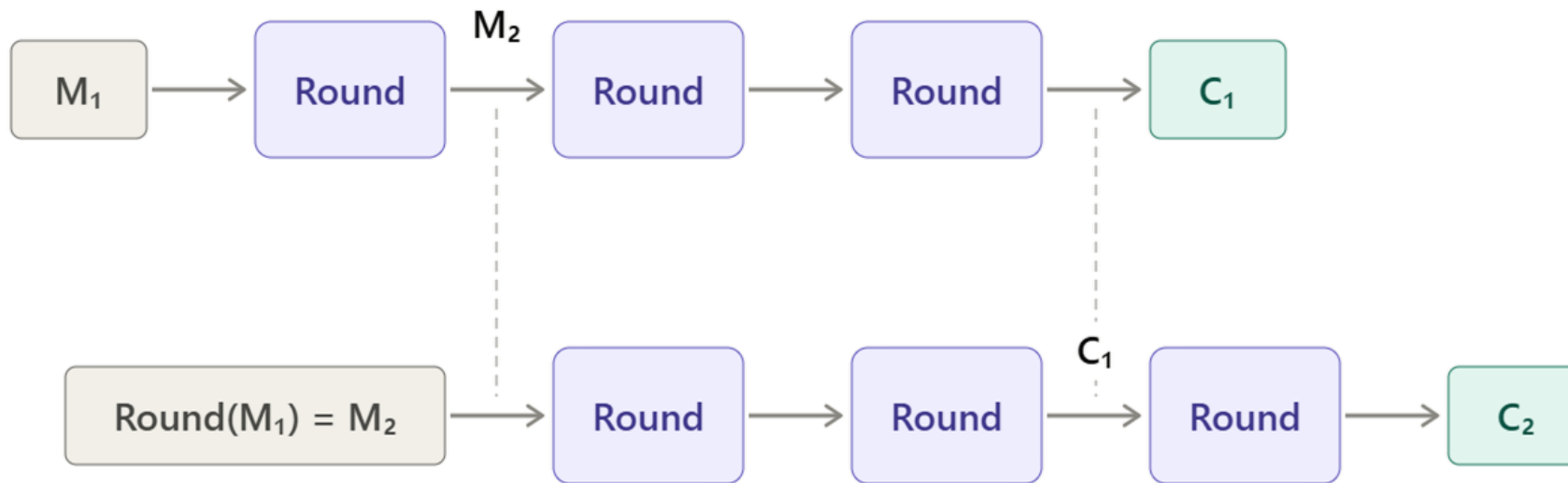
- **Σημείωση στην ορολογία:** Το block cipher δεν είναι κρυπτοσύστημα, αλλά συστατικό κρυπτοσυστήματος
- Γίνεται κρυπτοσύστημα αν συνδυάσουμε το block cipher διαδοχικά σε πολλά block ενός πλήρους μηνύματος με διάφορους τρόπους (***mode of operation***).
- Ανεξάρτητος σχεδιασμός block cipher και κρυπτοσυστήματος
- Διαφορετικές ιδιότητες ασφάλειας:
- Μοντελοποίηση ασφάλειας block cipher
 - Ψευδοτυχειότητα
 - Ιδιότητες PRP
- Μοντελοποίηση ασφάλειας κρυπτοσυστήματος
 - Ιδιότητες IND-CPA, IND-CCA κλπ.

Block Ciphers (Κρυπτοσυστήματα Block)

- Κρυπτογράφηση σε γύρους (rounds)
 - Ίδια λειτουργία $c = Enc_3(Enc_2(Enc_1(m)))$
 - Παραμετροποίηση με round key
 - **Key expansion:** δημιουργία round key k_i από το κλειδί k
 - Περισσότεροι γύροι – μεγαλύτερη ασφάλεια
- Βασική ιδέα γύρου (Shannon)
 - **Diffusion**
 - Ένα bit μηνύματος πρέπει να επηρεάζει πολλά bits κρυπτοκειμένου.
 - **Confusion**
 - Η σχέση μεταξύ των bits του κρυπτοκειμένου και των bits του κλειδιού πρέπει να είναι σύνθετη.
 - Μη γραμμικότητα
- Αποκρυπτογράφηση πάλι σε γύρους
 - Round keys σε αντίστροφη σειρά $m = Dec_1(Dec_2(Dec_3(m)))$
- Υλοποίηση:
 - Δίκτυα Feistel – S Boxes (DES)
 - Δίκτυα Substitution Permutation (AES)

Slide attack

- Γιατί οι γύροι πρέπει να είναι διαφορετικοί;
- Αν $M_2 = \text{Round}(M_1)$ τότε και $C_2 = \text{Round}(C_1)$



Για το μέγεθος του block

- **Μεγάλο μέγεθος block**

- Μεγάλες απαιτήσεις μνήμης
- Ασυμβατότητα με καταχωρητές CPU

- **Μικρό μέγεθος block - Codebook attacks**

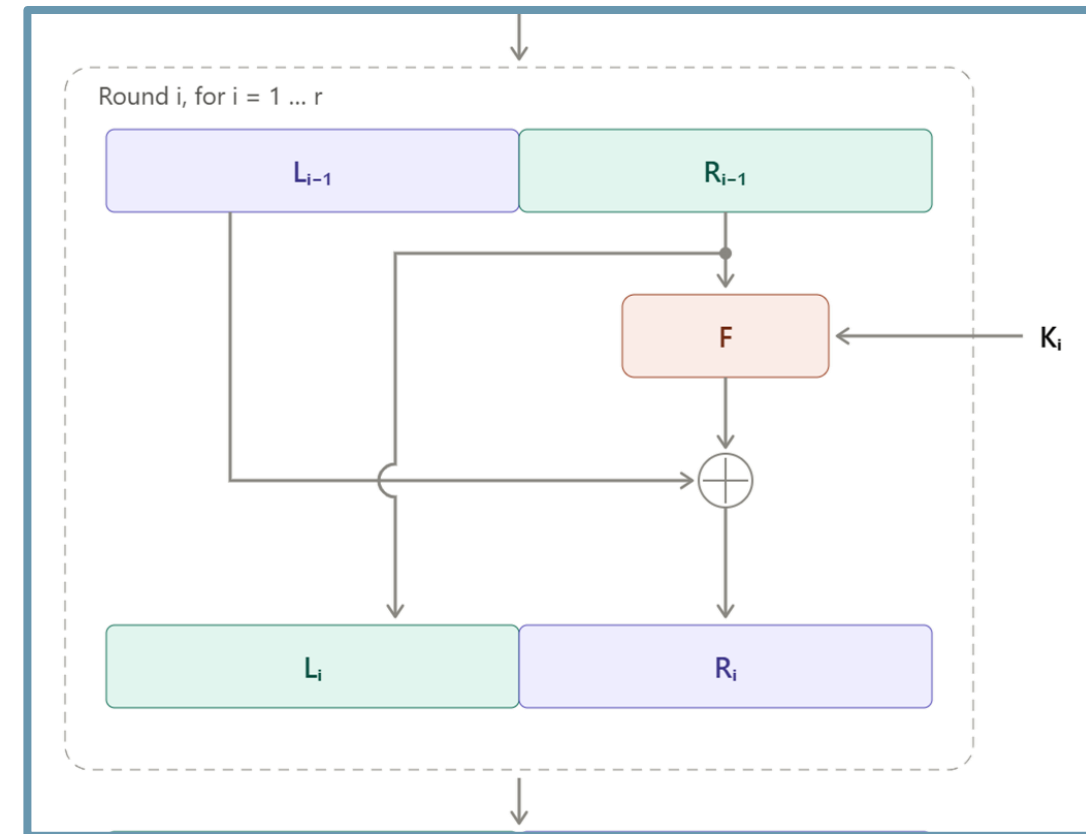
- Π.χ. Για block 16 bits
 - Δημιουργία όλων των 2^{16} πιθανών block ciphertexts
 - Chosen plaintext attack:
 - Κλήση encryption oracle με όλα τα πιθανά μπλοκ μηνύματος
 - Lookup table με 2^{16} καταχωρήσεις των 16 bits η καθεμία – σύνολο 2^{20} bits
- Για block 64 bits – μέγεθος lookup table 2^{70} bits.

Το κρυπτοσύστημα DES

- **Data Encryption Standard**
- Βασίζεται στα δίκτυα Feistel (H. Feistel - IBM - 1973)
- Πρότυπο FIPS (1977) – **Κλειστή διαδικασία (υπόνοιες για backdoors)**
 - Μέγεθος κλειδιού 56 *bits* (+8 parity bits)
 - Μέγεθος block 64 *bits*
 - 16 γύροι
 - Κλειδί γύρου 48 *bits*
- Από 90s θεωρούνται πολύ μικρές τιμές για παραμέτρους ασφάλειας
 - Σήμερα σπάνε σε ώρες
 - Δεν (πρέπει να) χρησιμοποιείται μόνο του

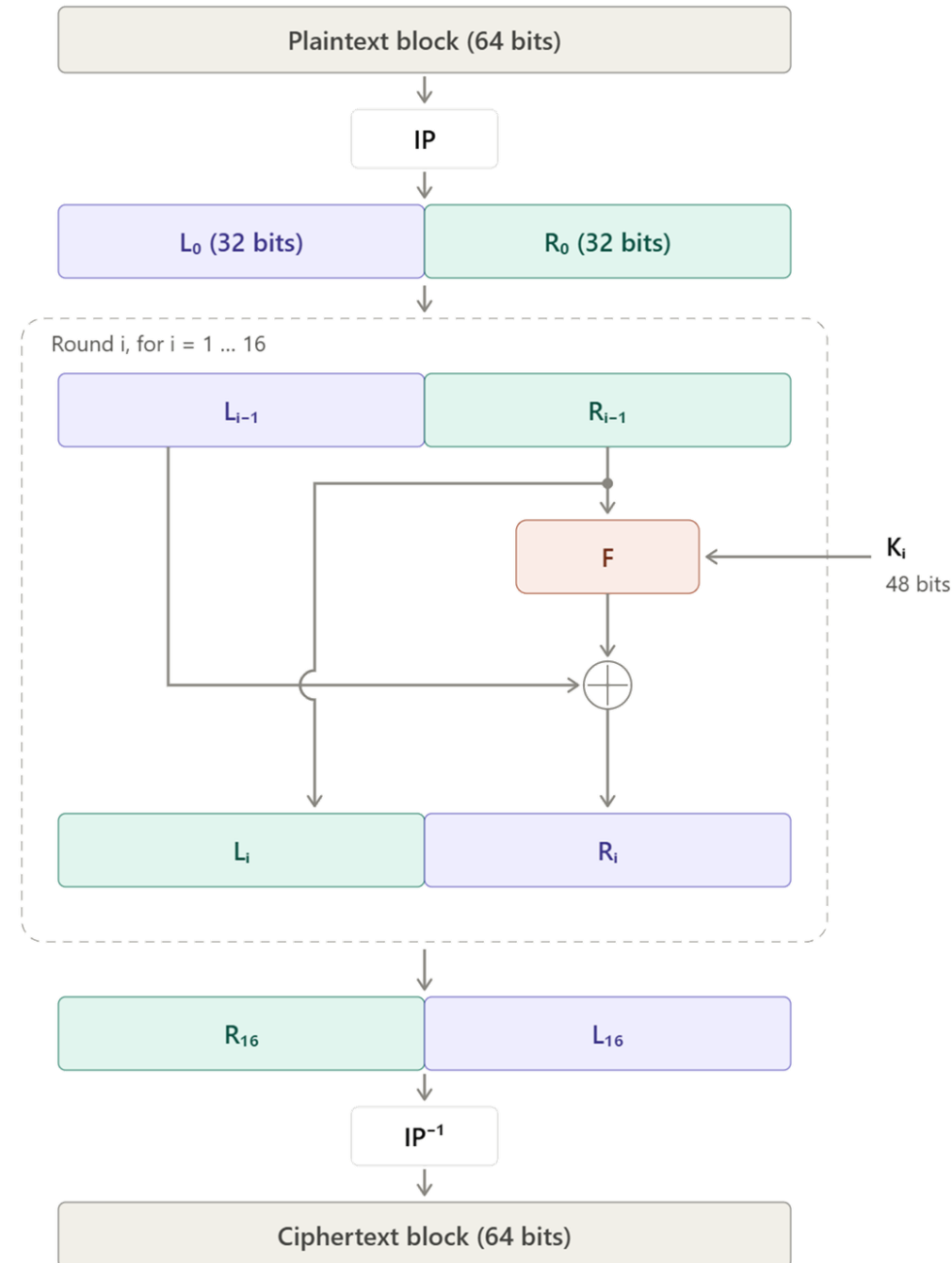
Δίκτυο Feistel – Γύρος

- Αναφερόμαστε σε block το οποίο σπάμε σε δύο κομμάτια
- Σε κάθε γύρο $i \geq 1$:
 - Είσοδος: $M_{i-1} = L_{i-1} || R_{i-1}$
 - $L_i \leftarrow R_{i-1}$
 - $R_i \leftarrow F(R_{i-1}, k_i) \oplus L_{i-1}$
 - k_i : κλειδί γύρου i – συνήθως προκύπτει από ολίσθηση του αρχικού κλειδιού
 - F : Μια συνάρτηση που προκαλεί confusion – diffusion
 - Δεν χρειάζεται να μπορεί να αντιστραφεί για την αποκρυπτογράφηση
 - $R_{i-1} \leftarrow L_i$
 - $L_{i-1} \leftarrow R_i \oplus F(R_{i-1}, k_i)$
 - Ίδια κυκλώματα – αντίστροφη σειρά

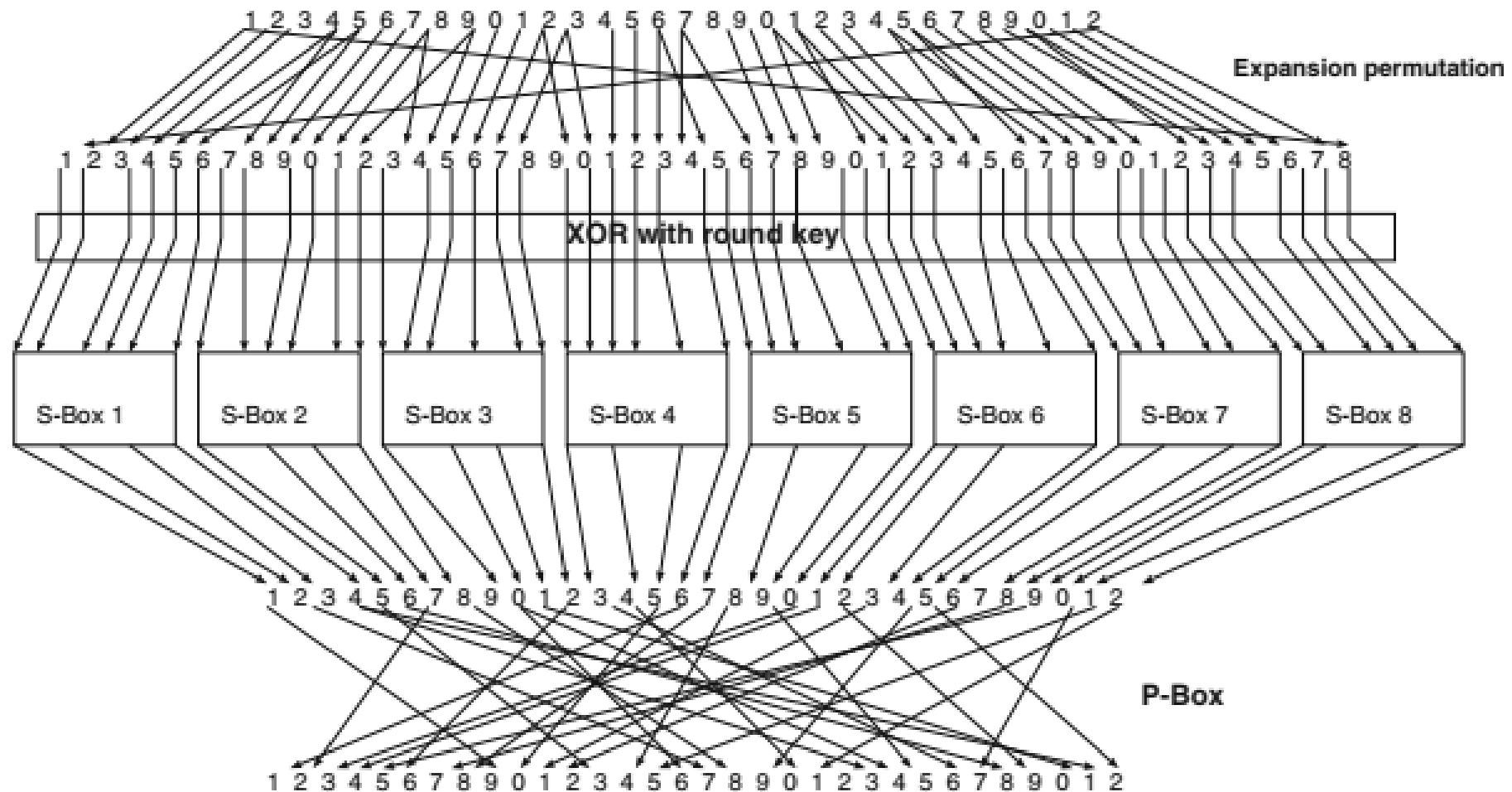


DES ως 16 γύροι Feistel

- Για κάθε μπλοκ:
 - Αρχική Μετάθεση IP.
 - Χωρισμός στα 2.
 - 16 εφαρμογές του γύρου Feistel διαλέγοντας 48bit κλειδιού
 - Ένωση των μπλοκ.
 - Αντίστροφη μετάθεση
- Οι αρχικές και τελικές μεταθέσεις δεν επηρεάζουν την ασφάλεια
 - Μάλλον για συμβατότητα με hardware εποχής (φόρτωση δεδομένων από 8bit buses)



Η ‘μαγική’ συνάρτηση F



Η ‘μαγική’ συνάρτηση F

- **Μετάθεση επέκτασης (expansion permutation) E**
 - $E: \{0,1\}^{32} \rightarrow \{0,1\}^{48}$
 - Μισό μπλοκ από 32 σε 46 bits.
 - Συμπληρώνει κάθε τετράδα bits μηνύματος εισόδου με ένα προηγούμενο και ένα επόμενο.
 - $E(b_0b_1b_2b_3) \rightarrow b_{31}b_0b_1b_2b_3b_4$
 - Από 8 τετράδες σε 8 εξάδες
- **Παραγωγή κλειδιού γύρου (48 bits)**
- **XOR με κλειδί γύρου**
 - Δεν χρησιμοποιείται αλλού
- **Συναρτήσεις αντικατάστασης S (S-Box)**
 - 8 συναρτήσεις με είσοδο μια εξάδα
 - Κάθε εξάδα αντικαθίσταται από μία τετράδα
- **Συνάρτηση μετάθεσης P (P-Box)**

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

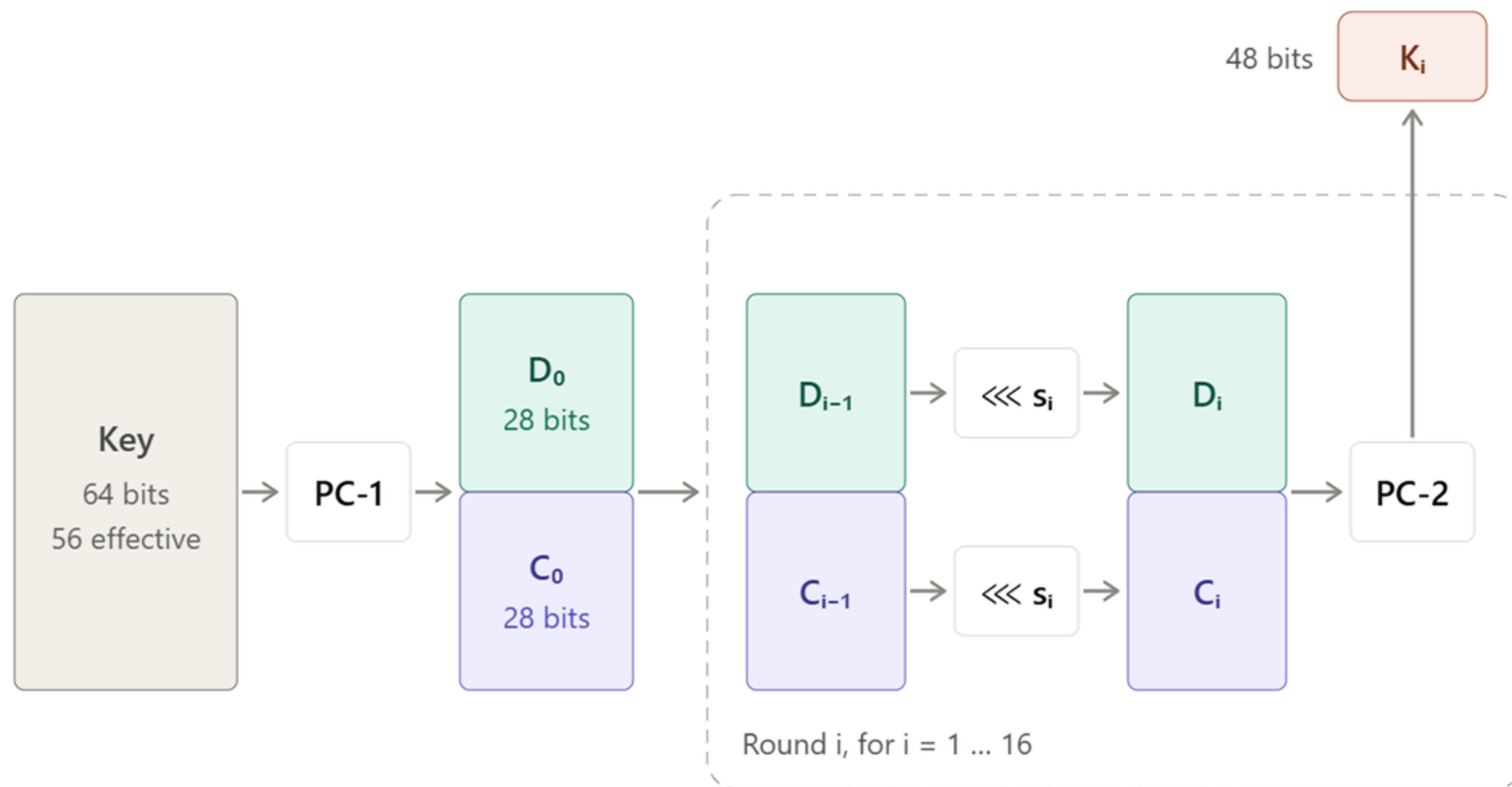
Παραγωγή κλειδιού γύρου – DES Key Schedule

Είσοδος: 64 bits κλειδιού

- Parity bits: κάθε 8άδα μονό αριθμό 1.
 - Αγνοούνται από το key schedule
- Εφαρμογή μετάθεσης PC1 στα υπόλοιπα $56 = 8 \times 7$
- Χωρισμός σε δύο μισά (28 bit) C_0, D_0
- Κυκλική Αριστερή Ολίσθηση κάθε μισού
 - $C_i \leftarrow C_{i-1} \ll s_i$
 - $D_i \leftarrow D_{i-1} \ll s_i$
 - $s_i = 1$, if $i \in \{1, 2, 9, 16\}$ else 2
- Εφαρμογή μετάθεσης PC2 σε $C_i || D_i$ αφαιρώντας 4 bits από κάθε μισό
 - 9, 18, 22, 25 (C)
 - 35, 38, 43, 54 (D)

57	49	41	33	25	17	9	PC1
1	58	50	42	34	26	18	
10	2	59	51	43	35	27	
19	11	3	60	52	44	36	
63	55	47	39	31	23	15	
7	62	54	46	38	30	22	
14	6	61	53	45	37	29	
21	13	5	28	20	12	4	PC2
14	17	11	24	1	5		
3	28	15	6	21	10		
23	19	12	4	26	8		
16	7	27	20	13	2		
41	52	31	37	47	55		
30	40	51	45	33	48		
44	49	39	56	34	53		
46	42	50	36	29	32		

Παραγωγή κλειδιού γύρου – DES Key Schedule



S-Boxes

Lookup table 4 γραμμών – 16 στηλών

- Περιέχει αριθμούς 4bits (0 – 15).
- Είσοδος 6 bits: $b_0b_1b_2b_3b_4b_5$
- Lookup τον αριθμό στην γραμμή b_0b_5 και στήλη $b_1b_2b_3b_4$
- Αποτέλεσμα το περιεχόμενο
 - $S[b_0b_5, b_1b_2b_3b_4]$

Για παράδειγμα:

- $S1[010101] \rightarrow 1100$
- $010101 \rightarrow (01, 1010) \rightarrow (1,10) \rightarrow 12 \rightarrow 1100$

S-Box 1															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S-Boxes

Ιδιότητες – Σχεδιαστικά κριτήρια

- Κάθε γραμμή είναι μετάθεση του $\{0, \dots, 15\}$.
- Κανένα **S-box** δεν είναι γραμμική συνάρτηση των εισόδων του.
- Αλλαγή ενός bit εισόδου επιφέρει αλλαγή σε τουλάχιστον δύο bit εξόδου
- Για κάθε bit εισόδου x_j και για κάθε bit εξόδου y_m ενός **S-box** ισχύει:
 - αν σταθεροποιήσουμε το x_j σε μια τιμή και αφήσουμε τα υπόλοιπα πέντε bit εισόδου να πάρουν όλους τους δυνατούς συνδυασμούς, τότε:
 - το y_m ισούται με 0 σε περίπου τις μισές από αυτές τις εισόδους και με 1 στις υπόλοιπες.

Κρυπτανάλυση Block Ciphers

- **Brute Force:**

- Μη πρακτικό αλλά γενικό.
- 2^{56} δοκιμές
- Complementarity property:
 - $DES(m, k) = c \Leftrightarrow DES(\bar{m}, \bar{k}) = \bar{c}$ (γιατί?)
 - Μειώνει τις brute force δοκιμές σε 2^{55}

- **Meet-In-The-Middle:**

- Πίνακες με Ενδιάμεσα Αποτελέσματα Κρυπτογράφησης – Αποκρυπτογράφησης Μηνυμάτων.

Κρυπτανάλυση Block Ciphers - Διαφορική Κρυπτανάλυση

- Ως αρχή ήταν γνωστή στην IBM από τη δεκαετία του 1970.
 - Αντιμετώπιση με S-Boxes.
- Όμως Biham & Shamir, 1990
 - $< 2^{50}$ δοκιμές με επιλεγμένα κρυπτοκείμενα
- Πώς συμπεριφέρεται το κρυπτοσύστημα στην διαφορά μεταξύ δύο μηνυμάτων;
 - Μεταφέρεται στα κρυπτοκείμενα;
- **Differential:** XOR δύο μηνυμάτων (κρυπτοκειμένων)
 - $\Delta_m = m_1 \oplus m_2$
 - $\Delta_c = Enc_k(m_1) \oplus Enc_k(m_2)$

- **Chosen Plaintext Attack**
 - Επιλέγω ζεύγη μηνυμάτων με συγκεκριμένο differential
 - Κοιτάω πώς αυτό μεταφράζεται στην έξοδο
- Αν το κρυπτοσύστημα είναι καλό, τότε κάθε differential της εξόδου θα είναι ισοπίθανο και ανεξάρτητο από το differential της εισόδου.

Κρυπτανάλυση Block Ciphers - Γραμμική Κρυπτανάλυση

- Στατιστική Επίθεση **KPA** - γνωστά κρυπτοκείμενα (μη επιλεγμένα)
 - Ανάκτηση κλειδιού DES με 2^{43} δοκιμές (Matsui, 1993)
- Βασική ιδέα: Προσέγγιση λειτουργίας του κρυπτοσυστήματος (ή τμήματος αυτού) με μια **γραμμική** σχέση.
 - Γραμμική σχέση: XOR (πρόσθεση *mod* 2)
 - $m_I \oplus c_J = 0$ με $I, J \subseteq [\text{block-length}]$
- Προσδιορισμός λειτουργιών του κρυπτοσυστήματος όπου τέτοιες σχέσεις ισχύουν με μεγάλη / μικρή πιθανότητα.
 - Αν υπάρχουν, σημαίνει ότι δεν λειτουργεί καλά – δείχνει την ύπαρξη δομής

Κρυπτανάλυση Block Ciphers - Γραμμική Κρυπτανάλυση

- **Bias** $\varepsilon = \left| \Pr[m_I \oplus c_J \oplus k_K = 0] - \frac{1}{2} \right|$ με $I, J \subseteq [\text{block-length}]$ και $K \subseteq [\lambda]$
- Το μόνο μη-γραμμικό συστατικό ενός block cipher είναι το S-box.
 - Όμως δεν είναι τέλεια μη-γραμμικό.
- Μπορεί να υπάρχουν συνδυασμοί εισόδων – εξόδων με μεγάλο bias.
- Καταγραφή τους για κάθε S-box
- Το bias μειώνεται κατά την λειτουργία του κρυπτοσυστήματος, όμως θα υπάρχει στο τέλος (σε κακή σχεδίαση).

Κρυπτανάλυση Block Ciphers - **Η επίθεση**

(και για differential και για linear cryptanalysis)

- Ξεκινάμε από το τέλος προσπαθώντας να μαντέψουμε τα bits του κλειδιού με βάση τα ζεύγη plaintext – ciphertexts.
- Αξιολόγηση επιλογής με βάση differential / μέγεθος bias.
- Differential cryptanalysis
χρειάζεται περισσότερα ζεύγη από linear cryptanalysis

Double DES

- Αποτυχημένη Προσπάθεια Αύξησης Ασφάλειας Με Διπλασιασμό του Κλειδιού
 - Κλειδί $k = k_1 || k_2$ (112 bits)
- Κρυπτογράφηση:
 - $\mathbf{DDES}(m, \mathbf{k}_1 || \mathbf{k}_2) \rightarrow \mathbf{DES}(\mathbf{DES}(m, \mathbf{k}_1), \mathbf{k}_2)$
- Αποκρυπτογράφηση:
 - $\mathbf{DDES}^{-1}(c, \mathbf{k}_1 || \mathbf{k}_2) \rightarrow \mathbf{DES}^{-1}(\mathbf{DES}^{-1}(c, \mathbf{k}_2), \mathbf{k}_1)$
- Ανάκτηση κλειδιού σε πολύ λιγότερο από 2^{112} προσπάθειες.
- Επίθεση Meet-In-The-Middle

• Double DES - Meet-In-The-Middle

- Με δεδομένο ένα ζεύγος μηνύματος, κρυπτοκειμένου (m, c)
- Δοκιμή όλων των 2^{56} κλειδιών για κρυπτογράφηση m .
- Αποθήκευση όλων των ενδιάμεσων κρυπτοκειμένων y .
- Δοκιμή όλων των 2^{56} κλειδιών για αποκρυπτογράφηση c .
- Αποθήκευση (ενδιάμεσων κρυπτοκειμένων).
- $\mathbf{DES}(m, k_1) \rightarrow y \leftarrow \mathbf{DES}^{-1}(c, k_2)$ και ανάκτηση k_1, k_2
- $2^{56} + 2^{56} = 2^{57}$ λειτουργίες DES και 2^{56} lookups σε $O(2^{56})$ space

• Double DES - Meet-In-The-Middle

- $\mathbf{DES}(\mathbf{m}, k_1), \mathbf{DES}^{-1}(\mathbf{c}, k_2)$
 - συνολικά 2^{112} ζευγάρια (πιθανά - η επίθεση δεν τα δοκιμάζει όλα)
- $\Pr[\mathbf{DES}(m, k_1) = \mathbf{DES}^{-1}(c, k_2)] = 2^{-64}$
- Συνολικά υπάρχουν $2^{112} \cdot 2^{-64} = 2^{48}$ συγκρούσεις
- Δοκιμή με δεύτερο ζεύγος μηνύματος, κρυπτοκειμένου (m', c')
 - Αν έχουν βρεθεί τα πραγματικά κλειδιά $\mathbf{DES}(m', k_1) = \mathbf{DES}^{-1}(c', k_2)$
 - Αναμενόμενος αριθμός νέων συγκρούσεων: $2^{48} \cdot 2^{-64} = 2^{-16}$

Triple DES

- Encrypt-Decrypt-Encrypt (EDE) paradigm
 - Με τρία διαφορετικά κλειδιά
 - $\text{TDES}(m, k_1 || k_2 || k_3) \rightarrow \text{DES}(\text{DES}^{-1}(\text{DES}(m, k_1), k_2), k_3)$ ή
 - Μέγεθος κλειδιού: *168 bits*
 - Στην πράξη χρειάζονται 2^{112} βήματα σε χρόνο και χώρο με chosen plaintext attack
 - Παραλλαγή με δύο κλειδιά
 - $\text{TDES}(m, k_1 || k_2) \rightarrow \text{DES}(\text{DES}^{-1}(\text{DES}(m, k_1), k_2), k_1)$
 - Μέγεθος κλειδιού: *112 bits*
 - Στην πράξη χρειάζονται 2^{56} βήματα σε χρόνο και χώρο με chosen plaintext attack

2-key Triple DES CPA Attack

- Για κάθε πιθανό κλειδί $k_i \in K$ υπολόγισε και αποθήκευσε το:
 - $\alpha_i \leftarrow \text{DES}_{k_i}^{-1}(0^{64})$ - κόστος $O(2^{56})$ σε χρόνο
 - Αποθήκευση των (α_i, k_i) σε πίνακα — κόστος $O(2^{56})$ σε χώρο
- Κάλεσε το Encryption Oracle για όλα τα α_i
 - $c_i \leftarrow \text{DES}_{k_1}(\text{DES}_{k_2}^{-1}(\text{DES}_{k_1}(\alpha_i)))$ - τα βλέπουμε ως plaintext
 - κόστος $O(2^{56})$ σε χρόνο, χώρο
- Υπολόγισε $b_i \leftarrow \text{DES}_{k_i}^{-1}(c_i)$ (με το αποθηκευμένο κλειδί k_i)
- Αναζήτησε στους πίνακες για ζεύγος (a_j, b_i) ώστε $a_j = b_i$
 - Με hashing ή sorting+binary search
- Αυτό θα είναι το πιθανό κλειδί $(k_i, k_j) = (k_1, k_2)$
- Δοκιμή με επιπλέον plaintexts για αντιμετώπιση συγκρούσεων

2-key Triple DES CPA Attack - Ορθότητα

$$\begin{aligned} b_i &= DES_{k_i}^{-1}(c_i) = DES_{k_i}^{-1} \left(DES_{k_i} \left(DES_{k_j}^{-1} \left(DES_{k_i}(\alpha_i) \right) \right) \right) && (k_1 = k_i, k_2 = k_j) \\ &= DES_{k_i}^{-1} \left(DES_{k_i} \left(DES_{k_j}^{-1} \left(DES_{k_i} \left(DES_{k_i}^{-1}(\mathbf{0}^{64}) \right) \right) \right) \right) && (\alpha_i = DES_{k_i}^{-1}(\mathbf{0}^{64})) \\ &= DES_{k_i}^{-1} \left(DES_{k_i} \left(DES_{k_j}^{-1} \left(DES_{k_i} \left(DES_{k_i}^{-1}(\mathbf{0}^{64}) \right) \right) \right) \right) && DES_{k_i} \left(DES_{k_i}^{-1}(\mathbf{0}^{64}) \right) = \mathbf{0}^{64} \\ &= DES_{k_i}^{-1} \left(DES_{k_i} \left(DES_{k_j}^{-1}(\mathbf{0}^{64}) \right) \right) = && \alpha_j = DES_{k_j}^{-1}(\mathbf{0}^{64}) \\ &DES_{k_i}^{-1} \left(DES_{k_i}(a_j) \right) = a_j \end{aligned}$$

AES

- Advanced Encryption Standard – NIST, 2000
- Αλγόριθμος Rijndael (Vincent **Rij**men - Joan **Da**emen)
- Μήκος block: *128 bits*
- Μήκος κλειδιού *128 bits = 16 bytes*
 - Το πλήθος των γύρων εξαρτάται από το μήκος του κλειδιού
 - *128 bits → 10 γύροι*
 - *192 bits → 12 γύροι*
 - *256 bits → 14 γύροι*
- Βασικό Συστατικό: Substitution Permutation Network
- Κάθε γύρος αποτελείται από:
 - Key addition
 - Substitution (Diffusion)
 - Permutation (Confusion – Non Linearity)
- Ισχυρή Μαθηματική Δομή
- Διαφορετική Κρυπτογράφηση - Αποκρυπτογράφηση

Περιγραφή του AES

- Αναπαράσταση: **Το μπλοκ μηνύματος (και το κλειδί)** αναπαρίσταται ως δισδιάστατος πίνακας *κατά στήλες* με:
 - κάθε στοιχείο είναι 1 byte
 - 4 γραμμές και τόσες στήλες ανάλογα με το μήκος του μπλοκ
- Για μπλοκ 128bits
$$\begin{pmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ b_{20} & b_{21} & b_{22} & b_{23} \\ b_{30} & b_{31} & b_{32} & b_{33} \end{pmatrix} = \begin{pmatrix} m_0 & m_4 & m_8 & m_{12} \\ m_1 & m_5 & m_9 & m_{13} \\ m_2 & m_6 & m_{10} & m_{14} \\ m_3 & m_7 & m_{11} & m_{15} \end{pmatrix}$$

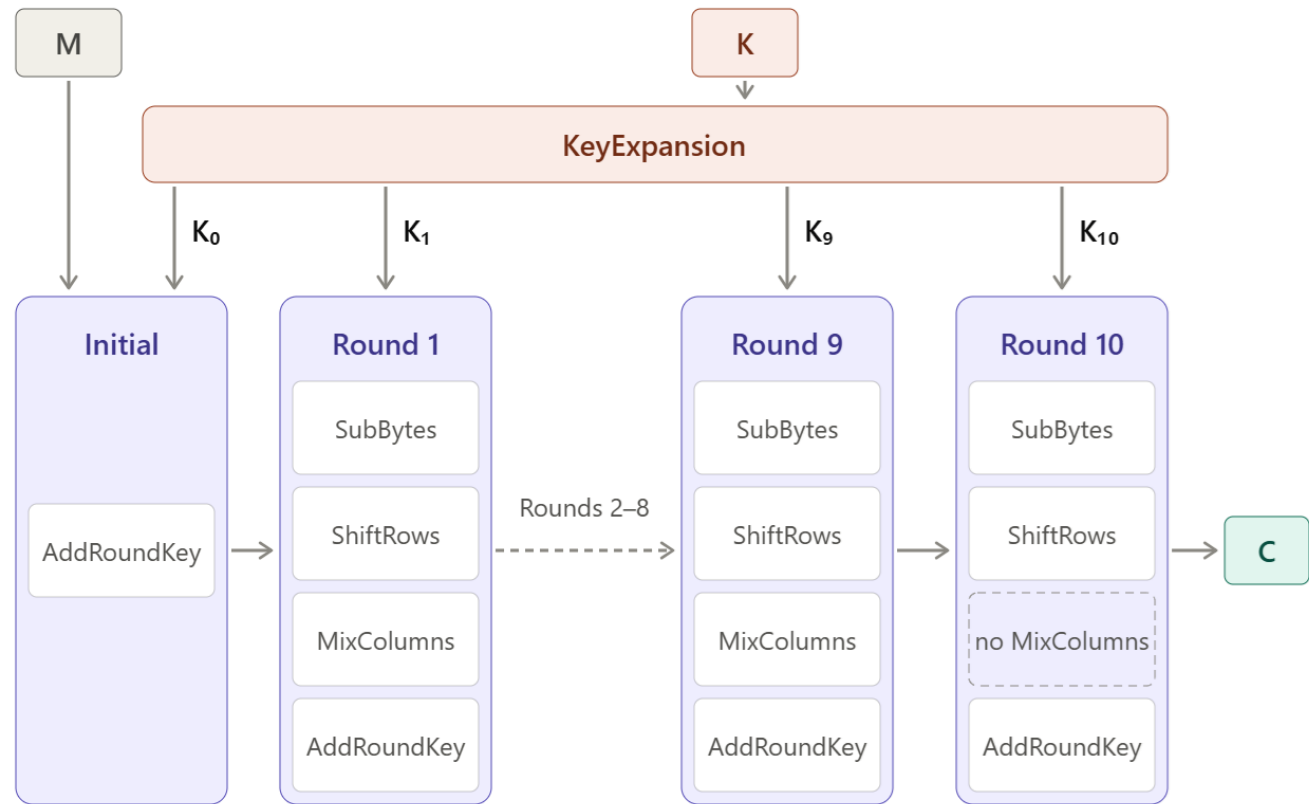
(w_0 w_1 w_2 w_3)
- Κάθε στήλη ονομάζεται λέξη (**word**)
- Ο πίνακας ονομάζεται **State**.
- Κάθε **γύρος** λαμβάνει ένα τέτοιο state, το τροποποιεί, και το επιστρέφει στην έξοδο.

High-Level AES Encryption

Σε κάθε γύρο εκτελούνται οι παρακάτω λειτουργίες:

- **AddRoundKey**
- **SubBytes**
- **ShiftRows**
- **MixColumns**

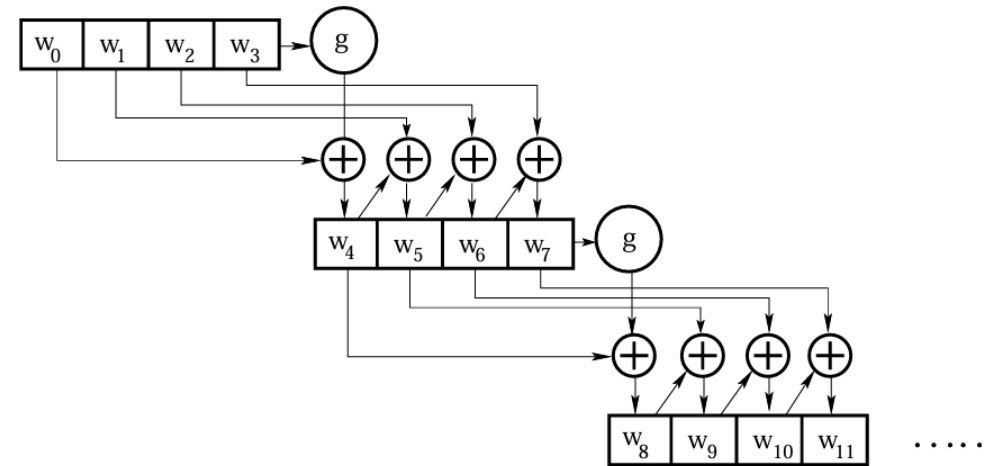
Στον τελευταίο γύρο αντί για την **MixColumns** εκτελούμε την **AddRoundKey**.



Για την αποκρυπτογράφηση εκτελούμε την **αντίστροφη** κάθε μιας από τις ενέργειες

AddRoundKey

- Εξαγωγή υποκλειδιού από το αρχικό κλειδί (key expansion)
 - Μετατροπή κλειδιού από 4 λέξεις σε 44.
 - Κάθε 4 λέξεις από την 5^η μέχρι και την 44^η χρησιμοποιούνται ως είσοδο σε κάθε γύρο.
 - **Συνάρτηση g**: Permutations
- Μετατροπή subkey σε 4×4 array από bytes
- $state' \leftarrow state \oplus subkey$
- Αν δεν υπήρχε οι γύροι θα ήταν ταυτόσημοι



ShiftRows

- Στόχος: Diffusion – Η αλλαγή σε μια στήλη πρέπει να επηρεάζει και τις υπόλοιπες
- Αλλαγή των bytes σε κάθε γραμμή του πίνακα state ως εξής:
 - 1^η γραμμή δεν αλλάζει
 - Κάθε byte στη **δεύτερη** γραμμή ολισθαίνει κυκλικά **μία** θέση αριστερά
 - Κάθε byte στην **τρίτη** γραμμή ολισθαίνει κυκλικά **δύο** θέσεις αριστερά
 - Κάθε byte στην **τέταρτη** γραμμή ολισθαίνει κυκλικά **τρεις** θέσεις αριστερά

$$\begin{pmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{11} & b_{12} & b_{13} & b_{10} \\ b_{22} & b_{23} & b_{20} & b_{21} \\ b_{33} & b_{30} & b_{31} & b_{32} \end{pmatrix} = \begin{pmatrix} m_0 & m_4 & m_8 & m_{12} \\ m_5 & m_9 & m_{13} & m_1 \\ m_{10} & m_{14} & m_2 & m_6 \\ m_{15} & m_3 & m_7 & m_{11} \end{pmatrix}$$

SubBytes / MixColumns

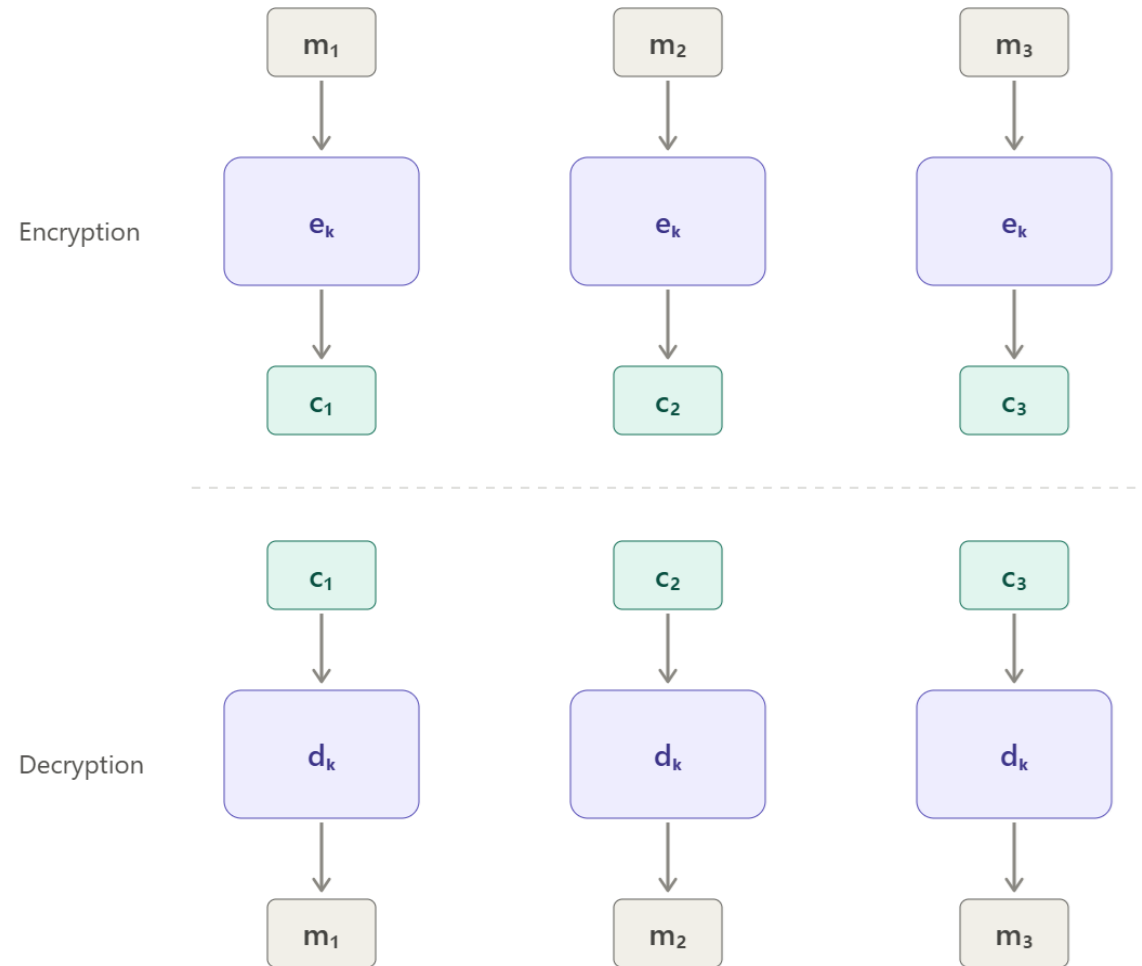
- Αντικατάσταση κάθε byte με διαφορετικό που προκύπτει από έναν πίνακα αντικατάστασης (S-box)
- S-box: Όλες οι δυνατές μεταθέσεις ενός byte ($|\{0,1\}^8| \rightarrow 256$)
- Το μη γραμμικό στοιχείο του AES
- Εφαρμογή αντιστρέψιμου γραμμικού μετασχηματισμού στα bytes κάθε λέξης (στήλης).
 - Αν δύο είσοδοι διαφέρουν κατά b bytes, τότε οι δύο έξοδοι θα διαφέρουν κατά $5 - b$ bytes

Modes of Operation

- Μετατροπή block ciphers σταθερού μεγέθους σε κρυπτοσυστήματα που μπορούν να χειριστούν μηνύματα μεταβλητού μεγέθους.
- Απαιτείται το block cipher να είναι *ενδομορφικό* δηλαδή plaintext space = ciphertext space.
- Υποθέτουμε ότι το μέγεθος του μηνύματος είναι πολλαπλάσιο του μεγέθους block - αν δεν είναι χρήση padding
- Ορίστηκαν αρχικά για το **DES**, αλλά έχουν επεκταθεί σε όλα τα block ciphers.

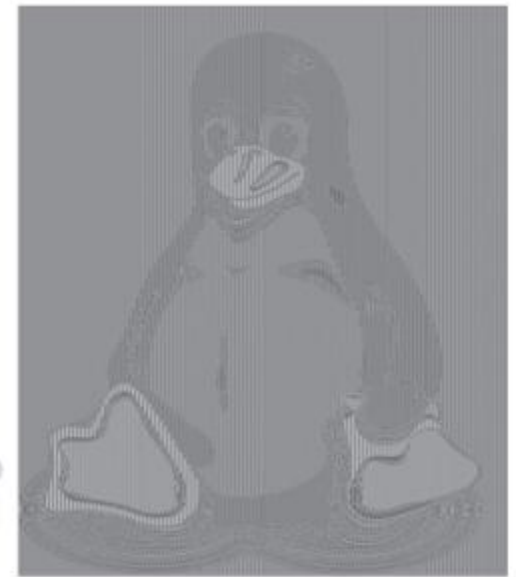
ECB (Electronic Code Book) Mode

- Κάθε block μηνύματος **κρυπτογραφείται χωριστά** και **ανεξάρτητα** με το ίδιο κλειδί
- Κάθε block κρυπτοκειμένου **αποκρυπτογραφείται χωριστά** και **ανεξάρτητα** με το ίδιο κλειδί



ECB (Electronic Code Book) Mode

- **Πλεονέκτημα:** Σε περίπτωση αλλοίωσης ενός μπλοκ, δεν επηρεάζονται τα υπόλοιπα.
- **Μειονέκτηματα:** Κάθε μπλοκ κρυπτογραφείται με τον ίδιο τρόπο.
 - Δύο ίδια μπλοκ θα δώσουν ακριβώς το ίδιο κρυπτοκείμενο.
 - **Είτε στο ίδιο, είτε σε διαφορετικά κρυπτοκείμενα!**
 - Ντετερμινιστική κρυπτογράφηση
 - Διατηρεί πιθανά μοτίβα που μπορεί να υπάρχουν στο μήνυμα.
 - Ευάλωτο σε στατιστικές επιθέσεις.
 - Δεν χρησιμοποιείται.



“Everybody knows ECB mode is bad because we can see the penguin”

ECB (Electronic Code Book) Mode

Θεώρημα: ECB δεν παρέχει ασφάλεια IND-CoA.

Θα εκτελέσουμε επιτυχή επίθεση χρησιμοποιώντας μηνύματα με δύο μπλοκ.

$$m_0 = \mathbf{0}||\mathbf{1} \text{ και } m_1 = \mathbf{1}||\mathbf{1}$$

Λαμβάνουμε μήνυμα c το οποίο σπάμε σε δύο μπλοκ:

Αν $c^{(0)} = c^{(1)}$ τότε επιστρέφουμε 1, αλλιώς επιστρέφουμε 0.

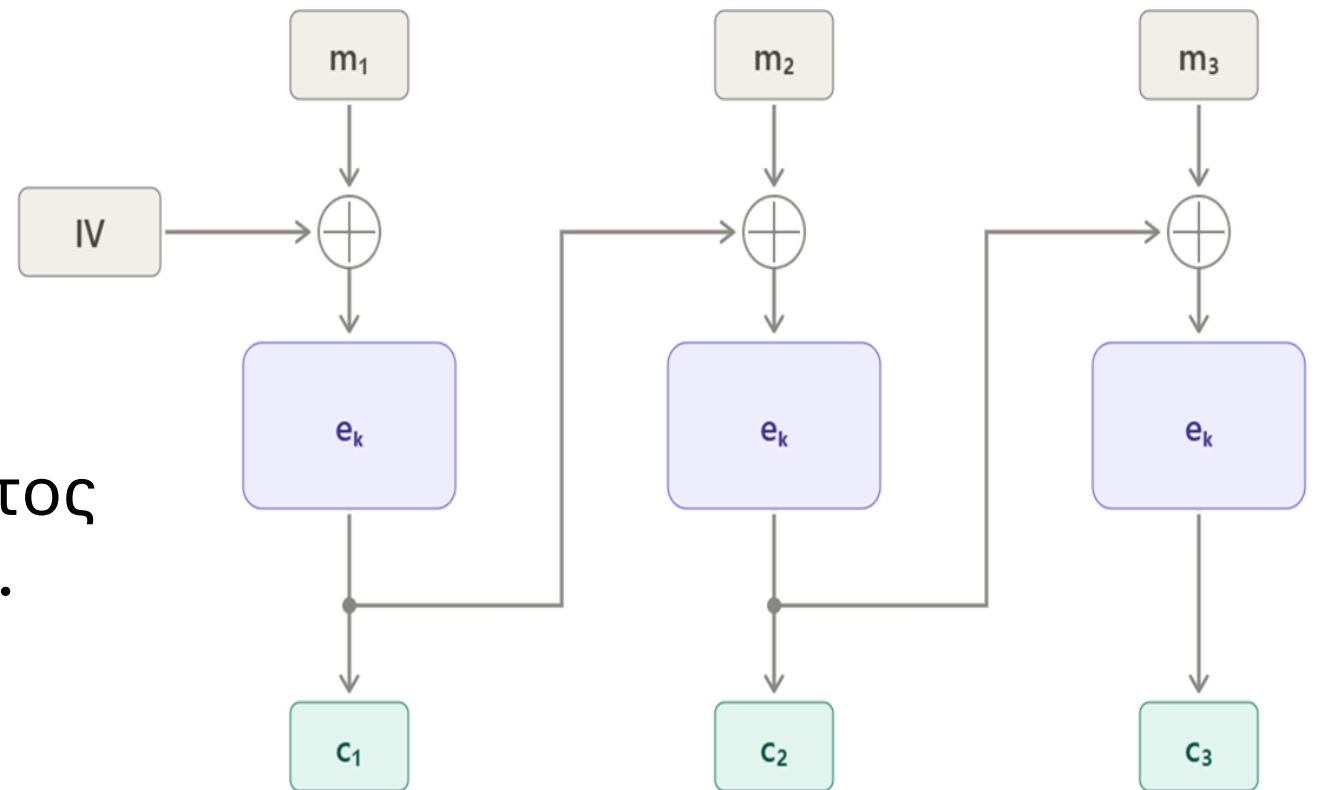
Θεώρημα: ECB δεν παρέχει ασφάλεια OW-CCA.

Θα εκτελέσουμε επιτυχή επίθεση για το c χρησιμοποιώντας μήνυμα με δύο μπλοκ: $c' = c||r$

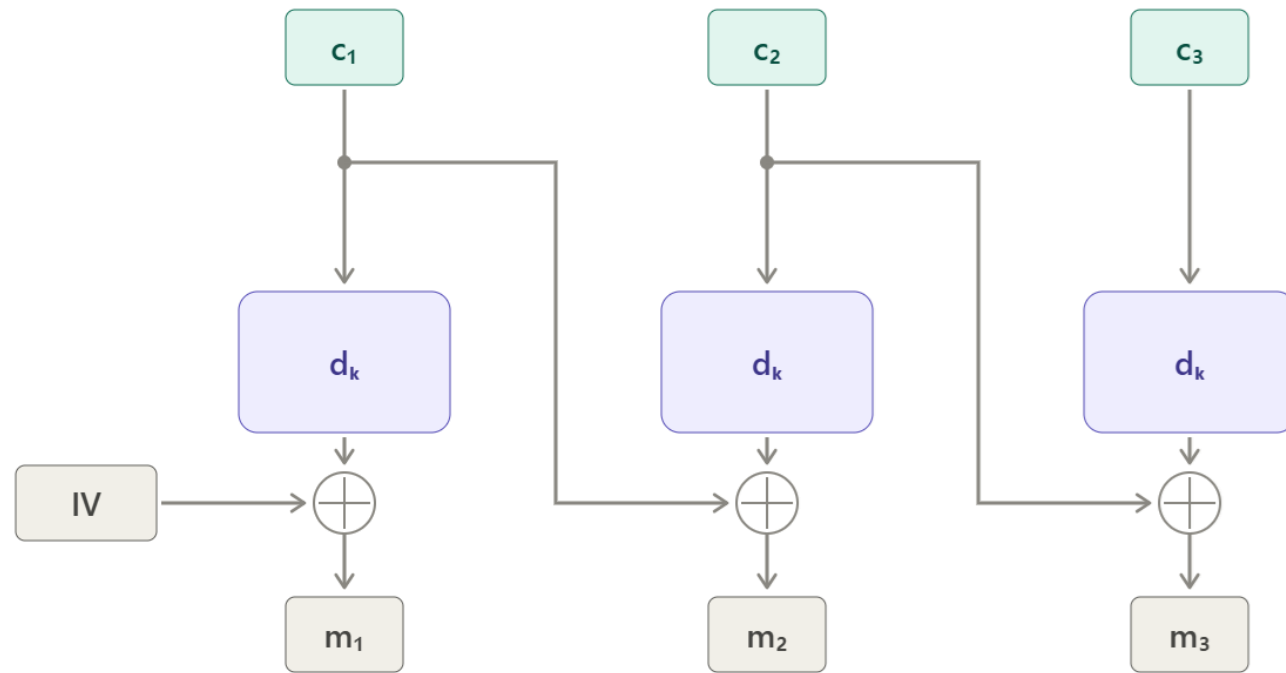
Λαμβάνουμε από το Μαντείο Αποκρυπτογράφησης μήνυμα m' το οποίο σπάμε σε δύο μπλοκ και επιστρέφουμε το πρώτο.

CBC (Cipher Block Chaining) Mode

- Κάθε μπλοκ μηνύματος συνδυάζεται XOR με το προηγούμενο μπλοκ κρυπτοκειμένου πριν κρυπτογραφηθεί.
 - $c_{i+1} \leftarrow Enc_k(m_{i+1} \oplus c_i)$
- Για το πρώτο μπλοκ μηνύματος χρησιμοποιείται τυχειότητα.
 - $c_0 \leftarrow IV$
 - Απαραίτητο για την αποκρυπτογράφηση



CBC (Cipher Block Chaining) Mode



- Πρώτα αποκρυπτογράφηση και μετά συνδυασμός με κρυπτοκείμενο προηγούμενου μπλοκ.

- $m_i \leftarrow Dec_k(c_i) \oplus c_{i-1}$

- $c_0 = IV$

- **Οι αποκρυπτογραφήσεις μπλοκ μπορούν να γίνουν παράλληλα!**

CBC (Cipher Block Chaining) Mode

- Δυνατότητες για τον χειρισμό του IV:
 - Ενσωμάτωση στο κρυπτοσύστημα
 - Σταθερή τιμή
 - Εσωτερική **τυχαία** επιλογή στον αλγόριθμο
 - **Nonce** που δίνεται ως είσοδος στην συνάρτηση κρυπτογράφησης
 - Πρόθεμα χωρίς κρυπτογράφηση στο κρυπτοκείμενο (η πιο συχνή χρήση του CBC mode)
 - Μοναδική τιμή (διαφορετικά **συνέπειες?**)
- Αλλοίωση ενός bit κρυπτοκειμένου αλλοιώνει ολόκληρο το μήνυμα
 - Χρήση για επαλήθευση ακεραιότητας (Message Authentication Code)

ECB (Electronic Code Book) Mode

Θεώρημα: CBC με σταθερό και nonce IV δεν παρέχει ασφάλεια IND-CPA.

Σταθερό IV: Ντετερμινιστικό κρυπτούστημα.

Nonce IV: Έστω nonce = $\mathbf{0}$.

Χρησιμοποιούμε το encryption oracle για το μήνυμα $(\mathbf{0}, \mathbf{0})$.

Παίρνουμε ως αποτέλεσμα το $(\mathbf{0}, c)$ με $c = \mathit{Enc}_k(\mathbf{0} \oplus \mathbf{0}) = \mathit{Enc}_k(\mathbf{0})$

Στην challenge φάση επιλέγουμε: $m_0 = (\mathbf{1}, \mathbf{0})$ και $m_1 = (\mathbf{1}, \mathbf{1})$

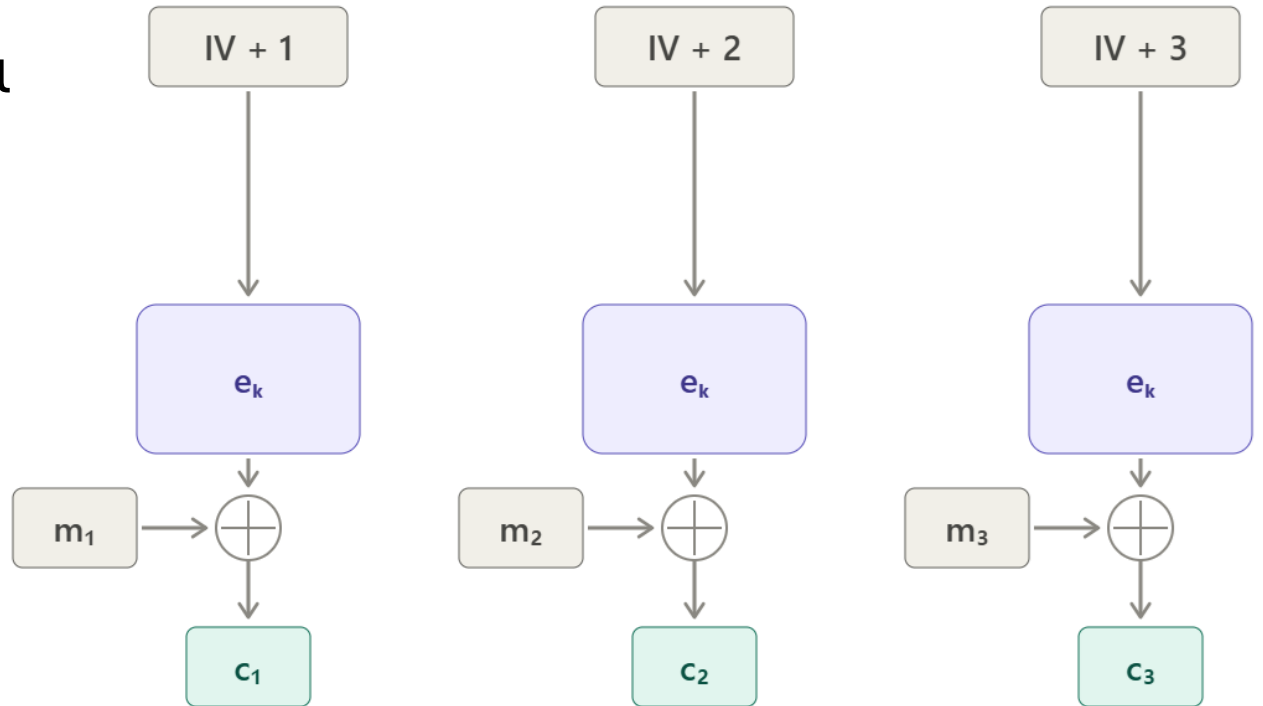
Παίρνουμε ως αποτέλεσμα το $(1, c^*)$ με

$$c^* = \mathit{Enc}_k\left(\mathbf{1} \oplus m_b^{(1)}\right) = \begin{cases} \mathit{Enc}_k(\mathbf{1}), & b = 0 \\ \mathit{Enc}_k(\mathbf{0}), & b = 1 \end{cases}$$

Επιστρέφουμε 1 αν $c^* = c$, αλλιώς 0.

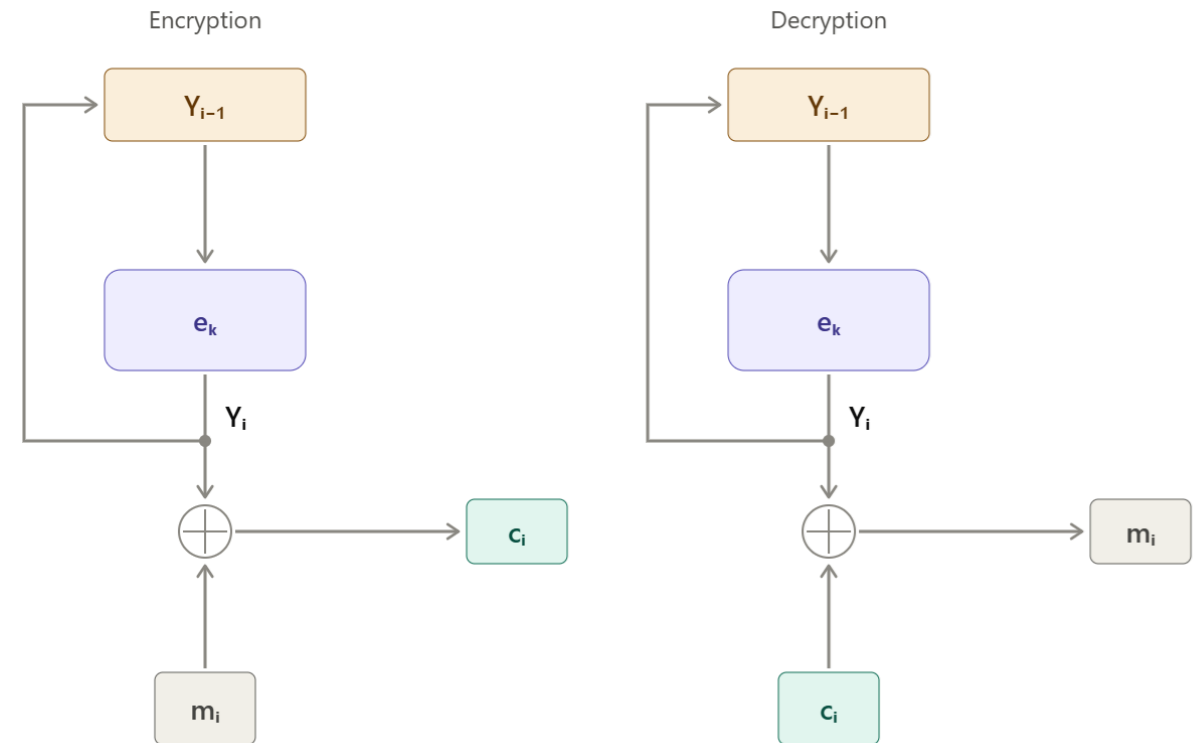
CTR (CounTeR) mode

- Κάθε block κρυπτοκειμένου παράγεται με XOR μηνύματος και νέου 'κλειδιού'
- Κλειδοροή (stream cipher)
- $c_i \leftarrow m_i \oplus Enc_k(IV + i)$
- $m_i \leftarrow c_i \oplus Enc_k(IV + i)$
- Ίδια κρυπτογράφηση / αποκρυπτογράφηση!
- **Παράλληλη επεξεργασία!**
- Αρκεί μοναδικό IV (όχι τυχαίο)



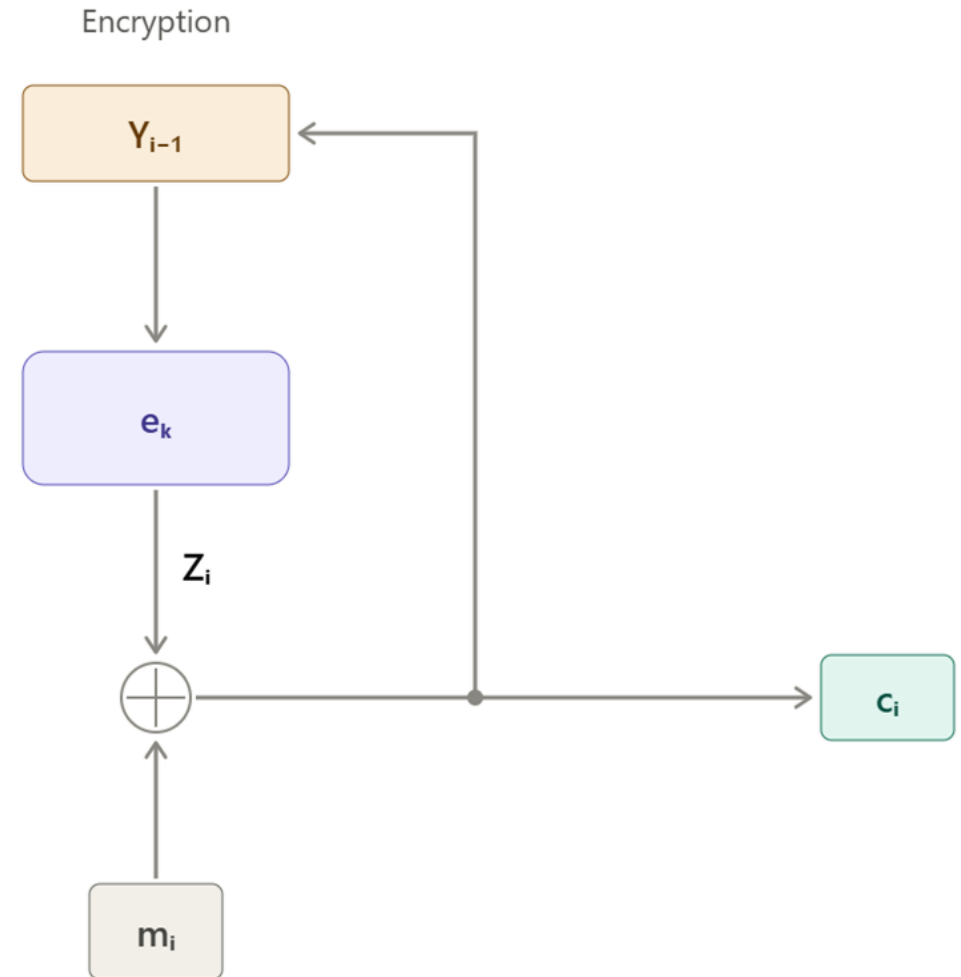
OFB (Output FeedBack) mode

- $y_i \leftarrow Enc_k(y_{i-1})$ και $y_0 \leftarrow IV$
- $c_i \leftarrow m_i \oplus y_i$
- Κλειδοροή ανεξάρτητη από το μήνυμα
- $y_i \leftarrow Enc_k(y_{i-1})$ και $y_0 \leftarrow IV$
- $m_i \leftarrow c_i \oplus y_i$



CFB (Cipher FeedBack) mode

- $c_i \leftarrow m_i \oplus Enc_k(c_{i-1})$ και $c_0 \leftarrow IV$
- Κλειδοροή εξαρτάται από το μήνυμα
- $m_i \leftarrow c_i \oplus Enc_k(c_{i-1})$ και $c_0 \leftarrow IV$



Message Padding

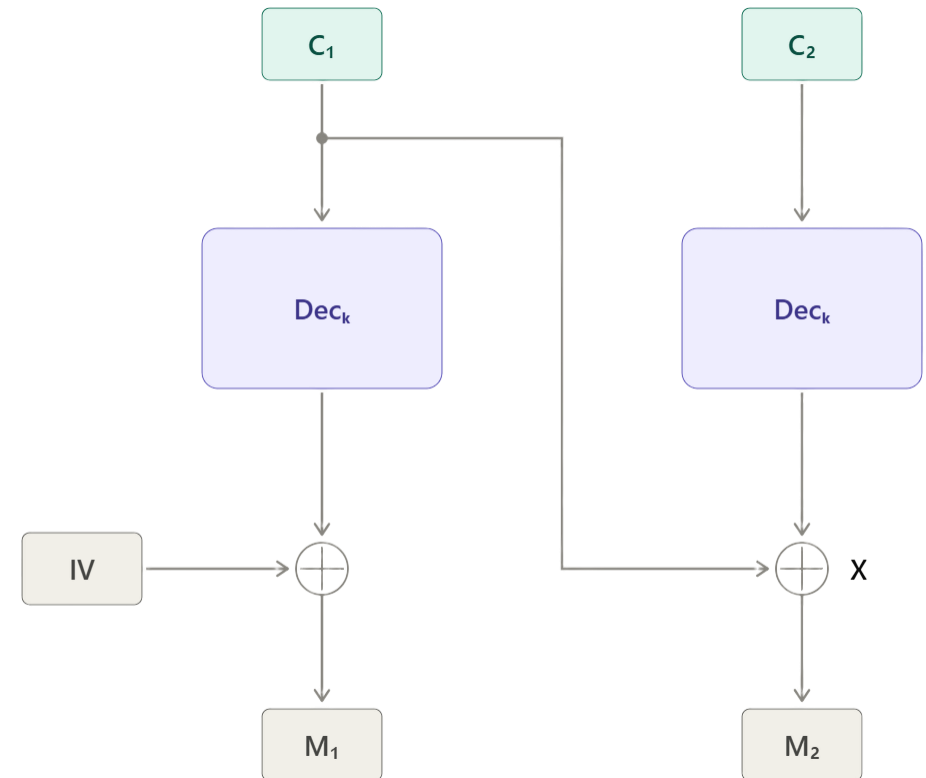
- Τι συμβαίνει όταν το μέγεθος του μηνύματος **ΔΕΝ** είναι πολλαπλάσιο του μεγέθους του block?
- Προσθήκη padding που περιέχει το πόσα bytes λείπουν.
- Για παράδειγμα, για μέγεθος block 16 bytes (128bits)
 - Αν λείπουν 15 bytes, προσθήκη 15bytes με περιεχόμενο 0f
 - Αν λείπουν 14 bytes, προσθήκη 14bytes με περιεχόμενο 0e
 - ...
 - Αν λείπουν 1 byte, προσθήκη 1byte με περιεχόμενο 01
 - Αν δε λείπει τίποτα προσθήκη 1 block με περιεχόμενο 10.
- Αποκρυπτογράφηση
- Έλεγχος ότι υπάρχει το σωστό padding
- Δηλαδή ότι το μήνυμα τελειώνει με:
 - 1byte με περιεχόμενο 01
 - 2byte με περιεχόμενο 02
 - ...
 - 15byte με περιεχόμενο 0f
 - 16byte με περιεχόμενο 10

Padding Oracle Attacks

- Επίθεση CCA
- Padding Oracle: Σύστημα το οποίο δίνει την πληροφορία για το αν το padding σε ένα μήνυμα είναι σωστό ή όχι
- Με βάση αυτό και αρκετά **επιλεγμένα ciphertexts** μπορούμε να αποκαλύψουμε το μήνυμα

Padding Oracle Attacks – Παράδειγμα CBC

- Έστω ότι έχουμε ένα μπλοκ C_2
- Θέλουμε να αποκαλύψουμε το $X = Dec_k(C_2)$
- Από CBC Mode: $X \oplus C_1 = M_2$
- Διαλέγουμε τυχαίο $\widehat{C}_1$ και σχηματίζουμε το ciphertext $\widehat{C}_1 || C_2$
- Θα αποκρυπτογραφηθεί μόνο αν $X \oplus \widehat{C}_1$ έχει έγκυρο padding.



Padding Oracle Attacks – Παράδειγμα CBC

- Δοκιμάζουμε διαδοχικά τιμές για το τελευταίο byte του $\widehat{C}_1$ μέχρις ότου:
 - $\widehat{C}_1[15] \oplus X[15] = 01$
- Με κατά μέσο όρο 128 δοκιμές θα βρεθεί το $X[15] = 01 \oplus C_1[15]$
- Θέσε $\widehat{C}_1[15] = X[15] \oplus 02$ ώστε $M_2[15] = 02$
- Αναζήτησε το $\widehat{C}_1[14]$ μέχρι να βρεθεί σωστό padding (0202) - $\widehat{C}_1[14] \oplus X[14] = 02$
- Επανάλαβε για όλα τα bytes.
- Με $128 * 16$ προσπάθειες κατά μέσο όρο μπορούν να βρεθούν όλα τα bytes του X.
- Στη συνέχεια: $M_2 = X \oplus C_1^*$ (το πραγματικό C_1)

