ΤΕΙ ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

ΑΝΟΙΚΤΑ ΑΚΑΔΗΜΑΙΚΑ ΜΑΘΗΜΑΤΑ

ΑΣΦΑΛΕΙΑ & ΔΙΑΧΕΙΡΙΣΗ ΔΙΚΤΥΩΝ(Θ)

Ενότητα 3: ΑΣΦΑΛΕΙΑ & ΔΙΑΧΕΙΡΙΣΗ ΔΙΚΤΥΩΝ

ΔΙΔΑΣΚΩΝ: ΚΩΝΣΤΑΝΤΙΝΟΣ ΧΕΙΛΑΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΤΕ

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

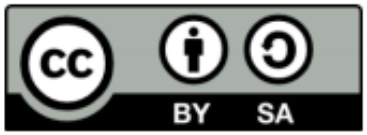
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

CC BY SA

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.

- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Κεντρικής Μακεδονίας» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Ενότητα 3

ΑΣΦΑΛΕΙΑ & ΔΙΑΧΕΙΡΙΣΗ ΔΙΚΤΥΩΝ

Χειλάς Κωνσταντίνος
Διδάκτορας Φυσικής

Περιεχόμενα ενότητας

1. Αλγόριθμοι συμμετρικού κλειδιού
2. P-box
3. S-Box
4. Product ciphers
5. Δομές τύπου Feistel
6. Παράμετροι και χαρακτηριστικά αλγορίθμων τύπου Feistel
7. DES – Data Encryption Standard
8. Χρόνος που απαιτείται για το σπάσιμο κώδικα ως συνάρτηση του μήκους κλειδιού
9. Triple-DES (3DES)
10. AES – Advanced Encryption Standard
11. *Rijndael (προφέρεται: rain-dahl*)
12. Ψευδοκώδικας υλοποίησης του *Rijndael* σε C
13. Blowfish
14. RC5
15. Αλγόριθμοι συμμετρικού κλειδιού
16. Τεχνικές κρυπτανάλυσης
17. Θέση λειτουργίας συσκευών κρυπτογράφησης

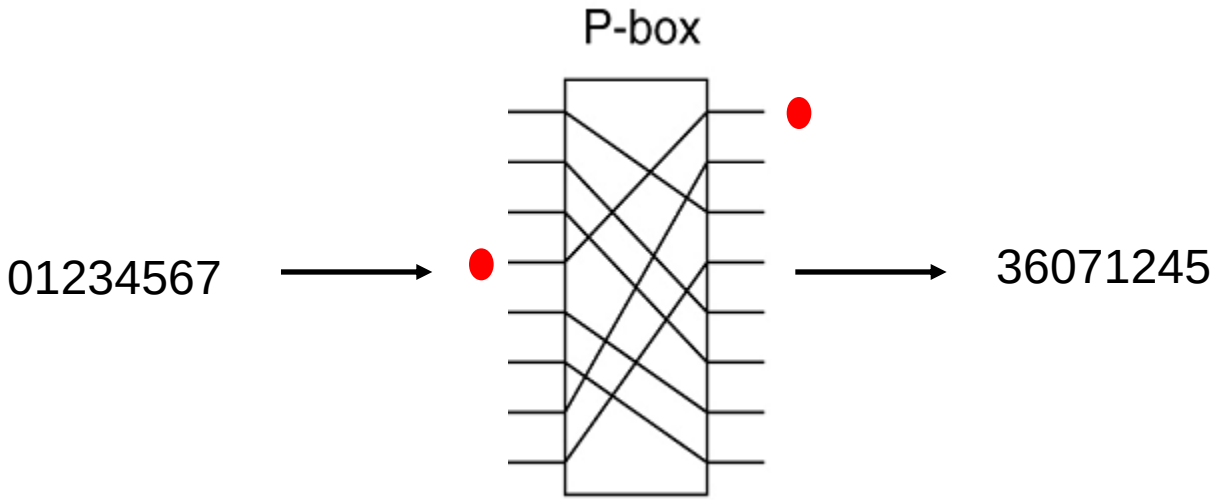
Σκοποί ενότητας

Αλγόριθμοι συμμετρικού κλειδιού

Αλγόριθμοι συμμετρικού κλειδιού

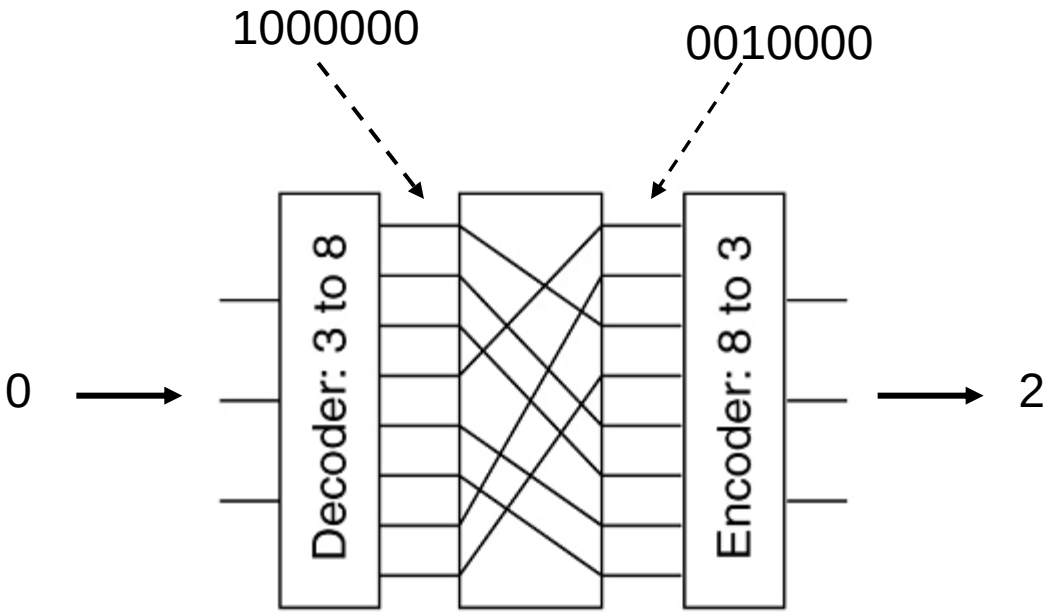
- Χρησιμοποιούν το ίδιο κλειδί για την κρυπτογράφηση και την αποκρυπτογράφηση
- Υλοποιούνται τόσο με υλικό (hardware) όσο και με λογισμικό (software)
- Hardware υλοποιήσεις:
  - P-Box (permutation box), Transposition
  - S-Box, Substitution
  - Product Ciphers

P-box



Ό,τι εισέρχεται από κάποια θέση της εισόδου, εξέρχεται από κάποια άλλη θέση της εξόδου, επομένως έχω αντιμετάθεση.
π.χ. αν υπάρχει είσοδος στη θέση 3 της εισόδου, αυτό το 3 θα εξέλθει από τη θέση 0 της εξόδου (θεωρώ παράλληλη είσοδο – έξοδο ώστε να έχει νόημα η αντιμετάθεση.

S-Box

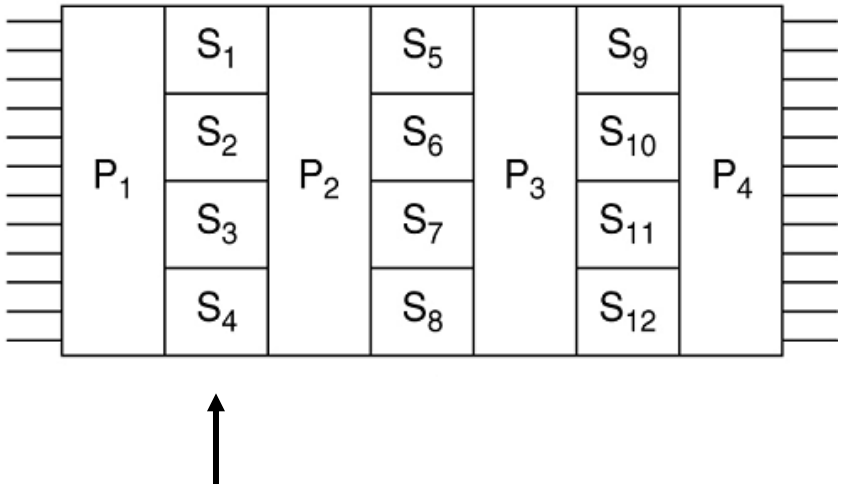


01234567 ⟶ ⟶ ⟶ 24506713

Κάθε χαρακτήρας που εισέρχεται από αριστερά
θα αντικατασταθεί από κάποιον στην έξοδο (δεξιά)

Product ciphers

- Λειτουργούν με k-bit εισόδους και k-bit εξόδους
- Τυπικά k=64 ή 256
- 18 επίπεδα (οι h/w υλοποιήσεις)
- >8 επαναλήψεις (rounds) οι s/w υλοποιήσεις



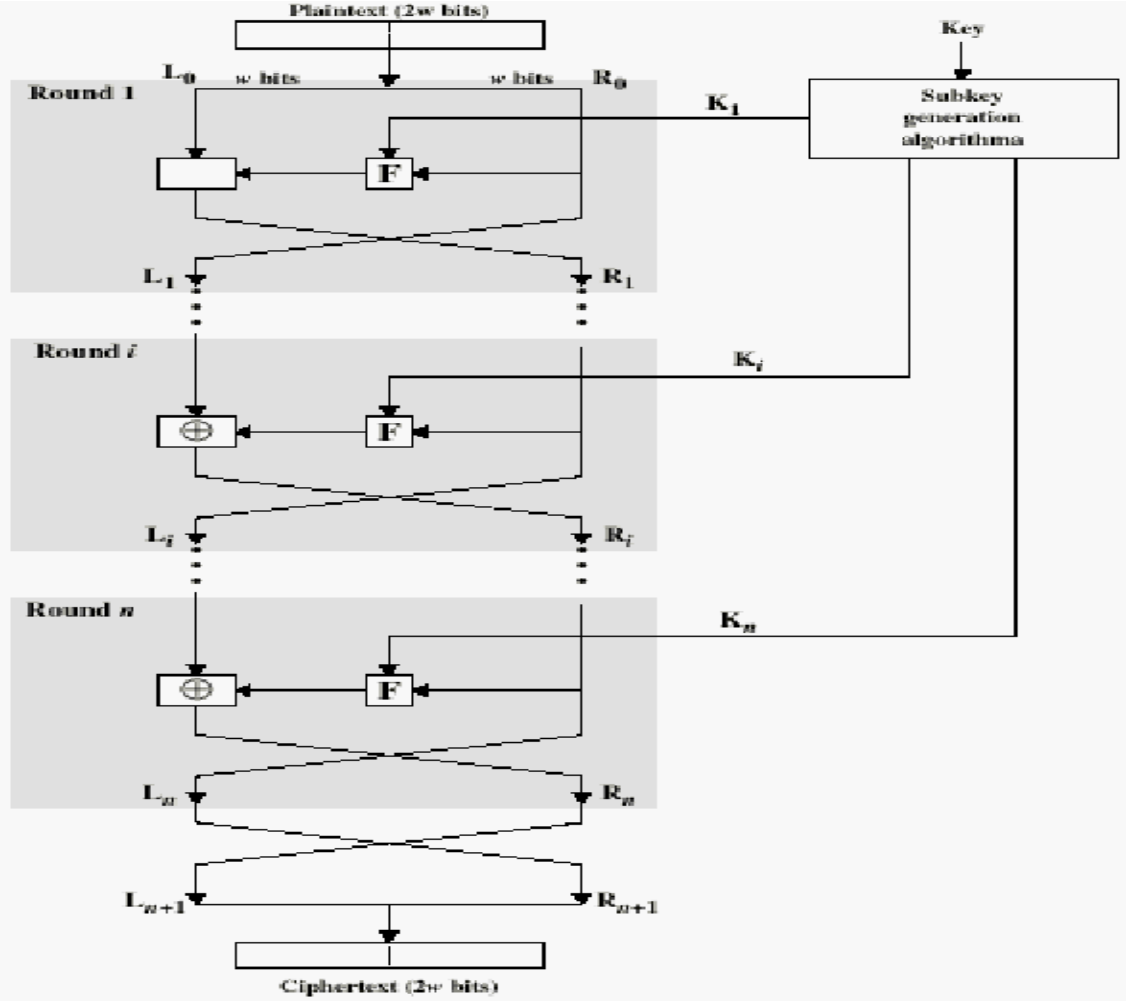
Αν ήταν ένα ενιαίο στάδιο
θα είχε 2^{12}=4096 καλώδια

Δομές τύπου Feistel

- Ουσιαστικά όλοι οι συμβατικοί αλγόριθμοι κρυπτογράφησης βασίζονται σε μια δομή που αρχικά παρουσιάστηκε από τον Horst Feistel της IBM το 1973
- Η επεξεργασία της πληροφορίας γίνεται σε κομμάτια ίσου μεγέθους (blocks) από το κείμενο, τα οποία παράγουν κρυπτογραφημένο κείμενο ίδιου μεγέθους
- Η υλοποίηση ενός δικτύου Feistel εξαρτάται από την επιλογή μιας από τις παρακάτω παραμέτρους και σχεδιαστικά χαρακτηριστικά:

Παράμετροι και χαρακτηριστικά αλγορίθμων τύπου Feistel

- Μέγεθος ομάδας (block): μεγαλύτερο μέγεθος block σημαίνει μεγαλύτερη ασφάλεια.
- Μέγεθος κλειδιού: μεγαλύτερο μέγεθος κλειδιού σημαίνει μεγαλύτερη ασφάλεια.
- Αριθμός επαναλήψεων (rounds): πολλαπλές επαναλήψεις οδηγούν σε μεγαλύτερη ασφάλεια
- Αλγόριθμος δημιουργίας υποκλειδιών (subkey generation algorithm): επηρεάζει την πολυπλοκότητα της κρυπτογράφησης κι επομένως δυσχεραίνει την κρυπτανάλυση.
- Γρήγορη κρυπτογράφηση και αποκρυπτογράφηση με λογισμικό: η ταχύτητα εκτέλεσης του αλγόριθμου είναι ένα σοβαρό σχεδιαστικό θέμα

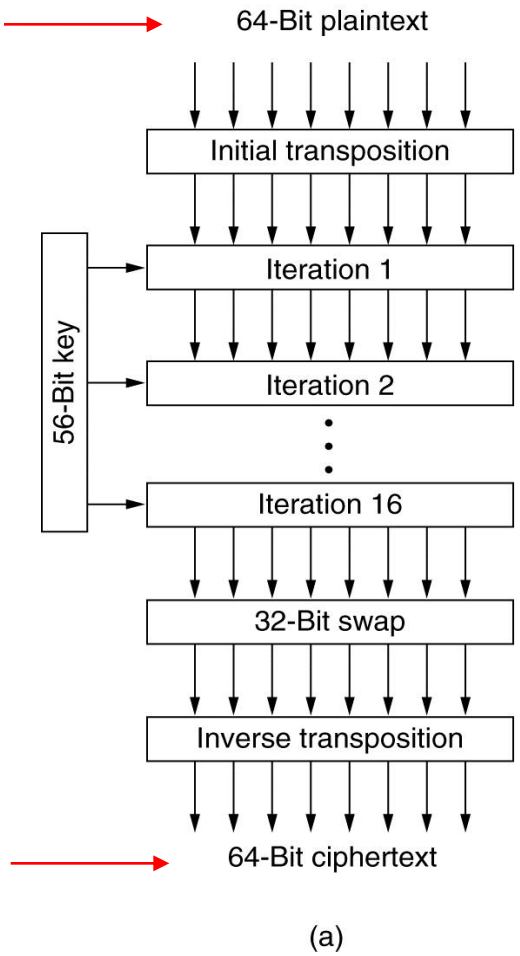

Plaintext (2w bits)
Key
Round 1
w bits
w bits
Subkey generation algorithma
F
Round i
Round n
Ciphertext (2w bits)

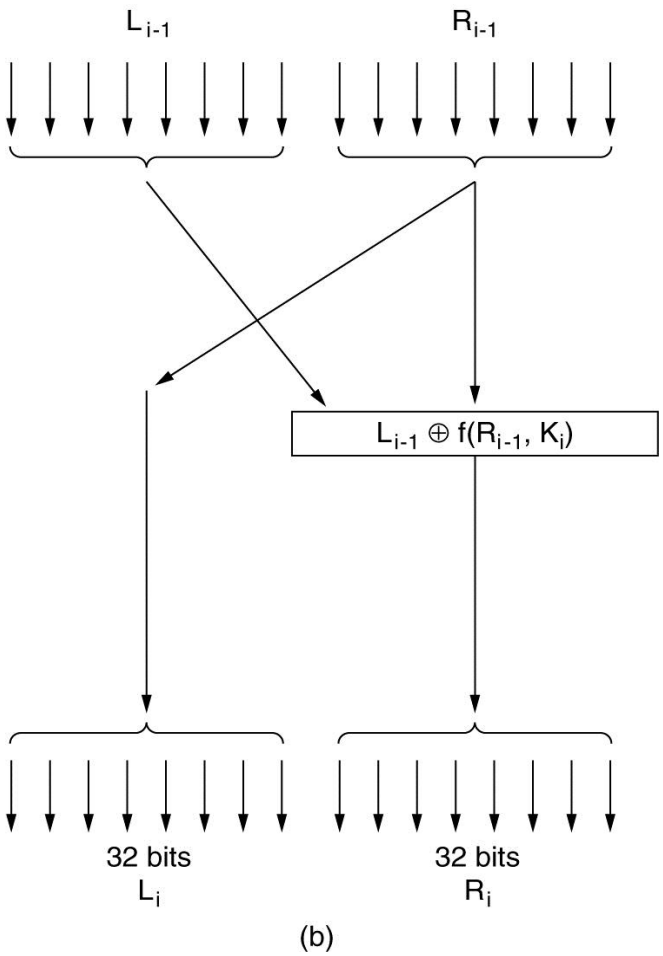
DES – Data Encryption Standard

- Αναπτύχθηκε από την IBM (Lucifer, 128-bit)
- Το 1977 η κυβέρνηση των ΗΠΑ το υιοθέτησε ως επίσημο πρότυπο κρυπτογράφησης για μη απόρρητες πληροφορίες (NSA, 56-bit)
- Η αρχική μορφή του αλγόριθμου δεν είναι πλέον ασφαλής
  - Diffie and Hellman (Stanford, 1977)
  - Μηχανή σπασίματος < 20M$ (σήμερα <200K$)
  - Exhaustive search – 2^{56} κλειδιά σε μια μέρα
- Χρησιμοποιούνται όμως κάποιες παραλλαγές του

DES

Πιθανό "whitening" πριν και μετά την εκτέλεση του αλγορίθμου με τη χρήση δύο κλειδιών 64-bit

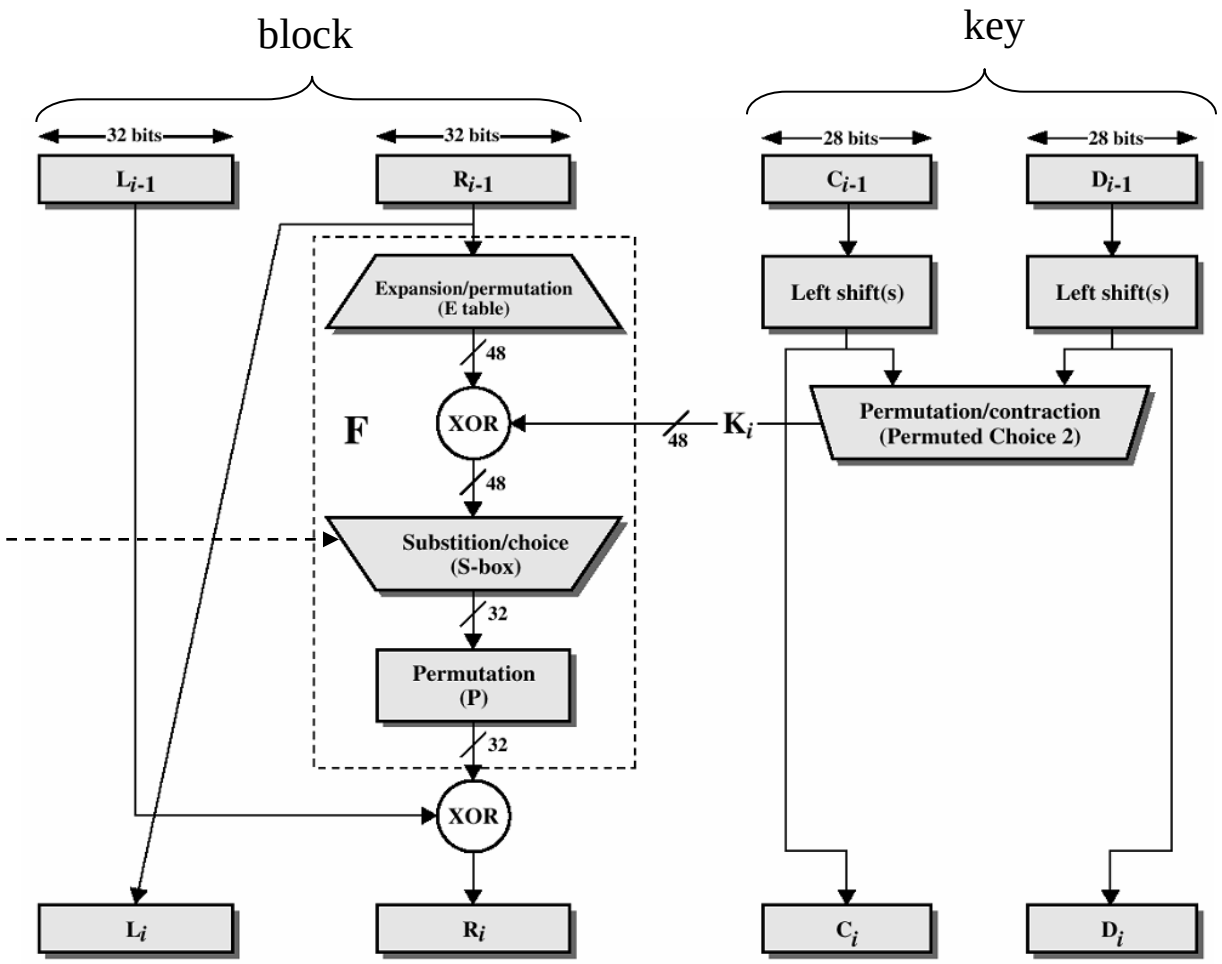




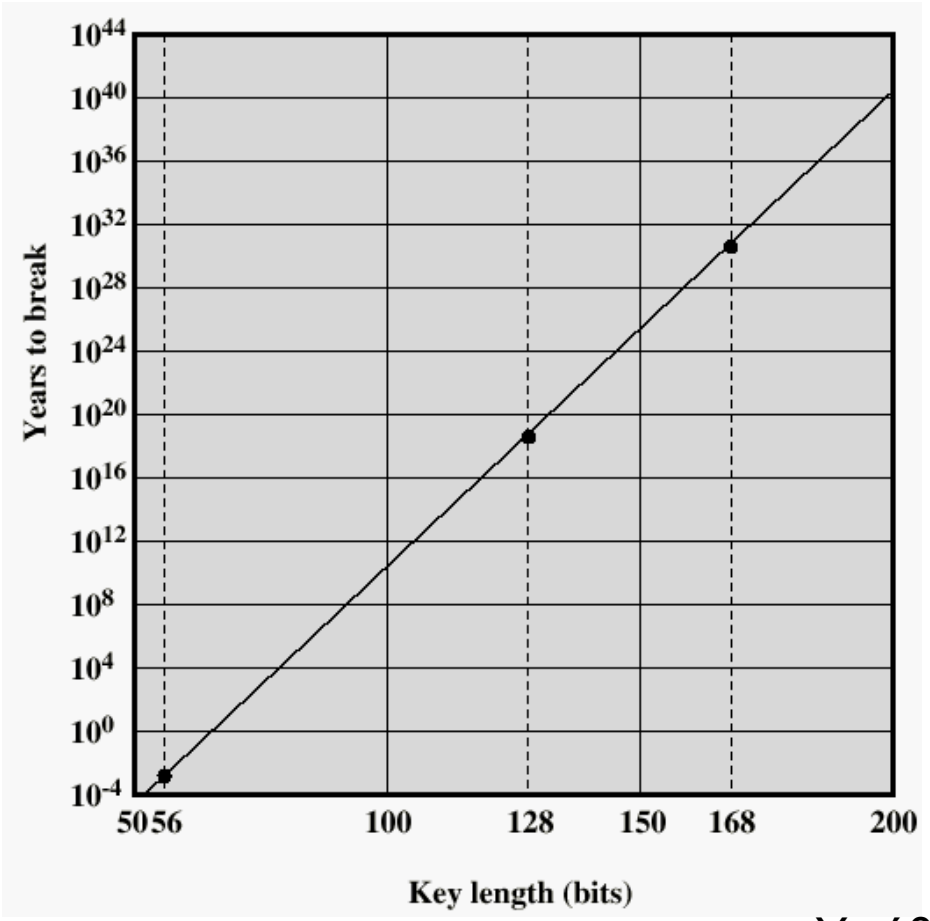
Λεπτομέρεια ενός κύκλου

Λειτουργία ενός κύκλου DES

Τα 48 bits σπάνε σε 8 X 6 bits και περνάνε από
8 διαφορετικά S-Boxes με 64 πιθανές εισόδους και 4 εξόδους, ώστε να προκύψουν 32 bits



Χρόνος που απαιτείται για το σπάσιμο κώδικα ως συνάρτηση του μήκους κλειδιού

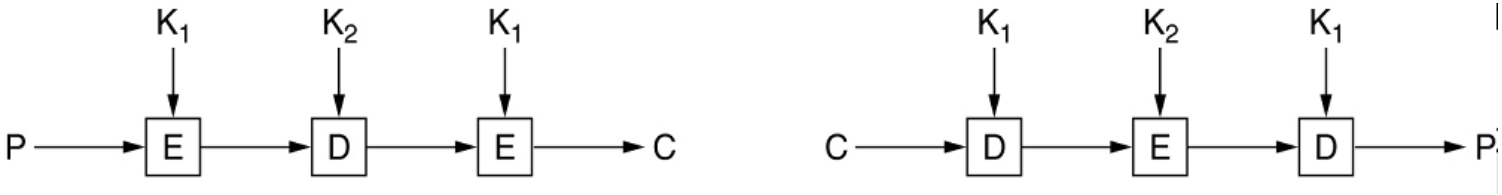


Υπόθεση 10^6 δοκιμές / μsec

Triple-DES (3DES)

- Triple Encryption (Tuchman, 1979)
- International Standard (IS) 8732
- 2 κλειδιά (k_1, k_2) και 3 στάδια
- Γιατί 2 κι όχι 3 κλειδιά;
  - Θεωρείται επαρκές: 112 bits → 2^{112} → 10^{14} χρόνια
- Γιατί EDE (Encrypt-Decrypt-Encrypt) και όχι EEE;
  - Συμβατότητα προς τα πίσω με το DES αν $k_1=k_2$.

Triple - DES



$$C = E_{K3}[D_{K2}[E_{K1}[P]]]$$

C = ciphertext
P = Plaintext
$E(K_i [X])$ = encryption of X using key K_i
$D(K_i [Y])$ = decryption of Y using key K_i

AES – Advanced Encryption Standard

- Το DES είναι σχεδιασμένο με βάση το h/w των 70's.
  - Οι υλοποιήσεις με λογισμικό είναι αργές
  - Block size = 64 bit ➔ χαμηλή απόδοση
- Το NIST έψαξε (Ιανουάριος 1997) τον αντικαταστάτη του με έναν ανοικτό
- διαγωνισμό κρυπτογραφίας

Όροι διαγωνισμού AES

1. Συμμετρικός block κώδικας
2. Ο σχεδιασμός να είναι δημόσιος
3. Υποστήριξη κλειδιών 128, 196 και 256 bits
4. Να μπορεί να υλοποιηθεί είτε με h/w είτε με s/w.
5. Ο αλγόριθμος πρέπει να διατίθεται δημόσια ή μέσω άδειας χωρίς όρους αποκλεισμού.

AES

- 15 υποψήφιοι αλγόριθμοι
- 1998: το NIST επέλεξε 5 υποψήφιους με βάση τα χαρακτηριστικά ασφάλειας, απόδοσης, απλότητας, ευελιξίας και απαιτήσεων μνήμης
- Μετά ακολούθησε δημόσια συζήτηση (συνέδρια, επιδείξεις, αναλύσεις, …) για τους πέντε επιλεγμένους αλγόριθμους
- Το τελευταίο συνέδριο ολοκληρώθηκε με μια ψηφοφορία:

AES

1. *Rijndael* (Joan Daemen & Vincent Rijmen, 86 ψήφοι)
2. *Serpent* (Ross Anderson, Eli Biham & Lars Knusden, 59 ψήφοι)
3. *TwoFish* (Bruce Schneier et al, 31 ψήφοι)
4. *RC6* (RSA Labs, 23 ψήφοι)
5. *MARS* (IBM, 13 ψήφοι)

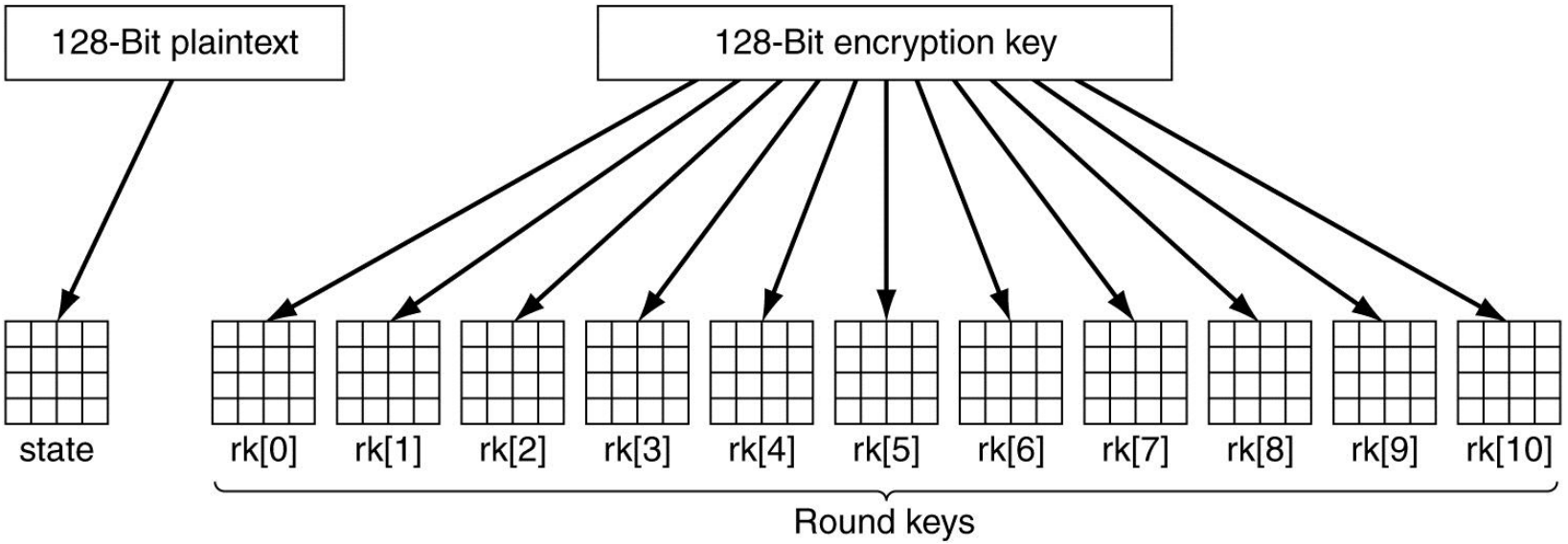
Οκτ. 2000 NIST επέλεξε τον *Rijndael* → 2001: FIPS 197 (Federal Information Processing Standard)

Rijndael *(προφέρεται: rain-dahl)*

- Υποστηρίζει κλειδιά και ομάδες κειμένου (block size) από 128 – 256 bits σε βήματα των 32 bit.
- Κύριες υλοποιήσεις:
  - block: 128 bits – key: 128 bit
  - block: 128 bits – key: 256 bit
- Η πρώτη δίνει: $2^{128} = 3 \text{ X } 10^{38}$ κλειδιά
- Αν είχα 1 δις παράλληλους επεξεργαστές που ο καθένας θα εκτελούσε 1 έλεγχο κλειδιού ανά 1 psec, θα χρειαζόμουν 10^{10} χρόνια για να ελέγξω όλο τον χώρο των πιθανών κλειδιών.

Rijndael

- Ο αλγόριθμος βασίζεται στην μαθηματική θεωρία των πεδίων Galois.



Δημιουργία των πινάκων state και rk

Ψευδοκώδικας υλοποίησης του *Rijndael* σε C

```
#define LENGTH 16                               /* # bytes in data block or key */
#define NROWS 4                                 /* number of rows in state */
#define NCOLS 4                                 /* number of columns in state */
#define ROUNDS 10                               /* number of iterations */
typedef unsigned char byte;                     /* unsigned 8-bit integer */

rijndael(byte plaintext[LENGTH], byte ciphertext[LENGTH], byte key[LENGTH])
{
  int r;                                        /* loop index */
  byte state[NROWS][NCOLS];                     /* current state */
  struct {byte k[NROWS][NCOLS];} rk[ROUNDS + 1];      /* round keys */

  expand_key(key, rk);                          /* construct the round keys */
  copy_plaintext_to_state(state, plaintext);    /* init current state */
  xor_roundkey_into_state(state, rk[0]);        /* XOR key into state */

  for (r = 1; r <= ROUNDS; r++) {
       substitute(state);                       /* apply S-box to each byte */
       rotate_rows(state);                      /* rotate row i by i bytes */
       if (r < ROUNDS) mix_columns(state);      /* mix function */
       xor_roundkey_into_state(state, rk[r]);   /* XOR key into state */
  }
  copy_state_to_ciphertext(ciphertext, state);  /* return result */
}
```

128 bits → 16 bytes

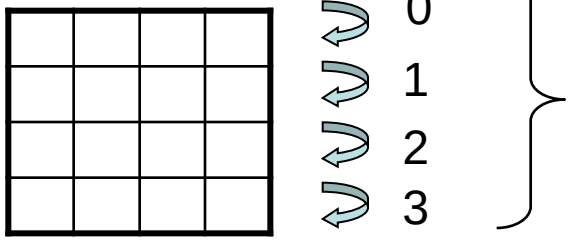
0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

state

x 10

state ⊕ rk[0]

1. S_0 → S-box → S_1
2. row rotation



S_2

3. Column mixing. Κάθε στήλη του S_2 πολλαπλασιάζεται με έναν σταθερό πίνακα χρησιμοποιώντας ιδιότητες πεπερασμένων πεδίων Galois, $GF(2^8)$ → S_3
(Daemen and Rijmen, 2002, Appendix E)
4. S_3 ⊕ rk[i] → S_4

Διαδικασία μετασχηματισμού του κάθε state (plaintext) σε ciphertext (S_4)

Blowfish

- 1993, Bruce Schneier
- Πολύ συμπαγής και γρήγορος αλγόριθμος. Απαιτεί μόλις 5K μνήμης
- Κλειδί 128 – 448 bits. Συνήθως 128 bits
- 16 rounds
- Δυναμικά S-Box (συναρτήσεις του κλειδιού), XOR και ADD
- Τα επιμέρους κλειδιά και τα S-boxes προκύπτουν με εφαρμογή του ίδιου του αλγόριθμου στο κλειδί. Αυτό απαιτεί 521 εκτελέσεις.
- Δεν είναι κατάλληλος για εφαρμογές στις οποίες το κλειδί πρέπει να αλλάζει συχνά.
- Δεν έχουν βρεθεί σοβαρές αδυναμίες
- Λόγω της αργής ταχύτητάς του, σήμερα χρησιμοποιείται ο TwoFish.

RC5

- 1994, Ron Rivest (RFC 2040) RSA Data Security Inc.
- Χαρακτηριστικά αλγορίθμου:
  1. Κατάλληλος για υλοποίηση με s/w και h/w
  2. Γρήγορος: δουλεύει πάνω σε λέξεις (word oriented)
  3. Το μέγεθος των λέξεων μπορεί να αλλάζει (προσαρμόζεται σε διαφορετικούς επεξεργαστές)
  4. Μεταβλητός αριθμός επαναλήψεων (rounds)
     1. Speed ⟷ Security
  5. Μεταβλητό μήκος κλειδιού
  6. Απλότητα
  7. Χαμηλές απαιτήσεις μνήμης
  8. Υψηλή ασφάλεια
  9. Οι κύκλοι εξαρτώνται από την ποσότητα δεδομένων

Αλγόριθμοι συμμετρικού κλειδιού

Αλγόριθμος	**Συγγραφέας**	**Κλειδί**	**Σχόλια**
Blowfish	Br. Schneier	1 – 288 bits	Παλιός και αργός
DES	IBM	56 bits	Σήμερα αδύναμος
IDEA	Masse & Xuejia	128 bits	Καλός αλλά πατενταρισμένος
RC4	R. Rivest	1- 2048 bits	Μερικά κλειδιά είναι αδύναμα
RC5	R. Rivest	128 – 256 bits	Καλός αλλά πατενταρισμένος
Rijndael	Daemen & Rijmen	128 – 256 bits	Καλύτερη επιλογή
Serpent	Anderson, Biham, Knusden	128 – 256 bits	Πολύ ισχυρός
3DES	IBM	168 bits	Δεύτερη καλύτερη επιλογή
TwoFish	Br. Schneier	128 – 256 bits	Πολύ ισχυρός και διαδεδομένος

πηγή: Computer Networks, Andrew Tanenbaum,4th ed., 2003

P4 2.1 GHZ CPU running Windows XP SP1

Algorithm	Data	Time (In Seconds)	Average MB/Second (approx.)	Performance
DES	256 MB	10 - 11	22 - 23	Low
3DES	256 MB	12	12	Low
AES (256-bit)	256 MB	5	51.2	Medium
Blowfish	256 MB	3.5 - 4	64	High

Πηγή: http://www.brighthub.com/computing/smb-security/articles/75099.aspx , 2010

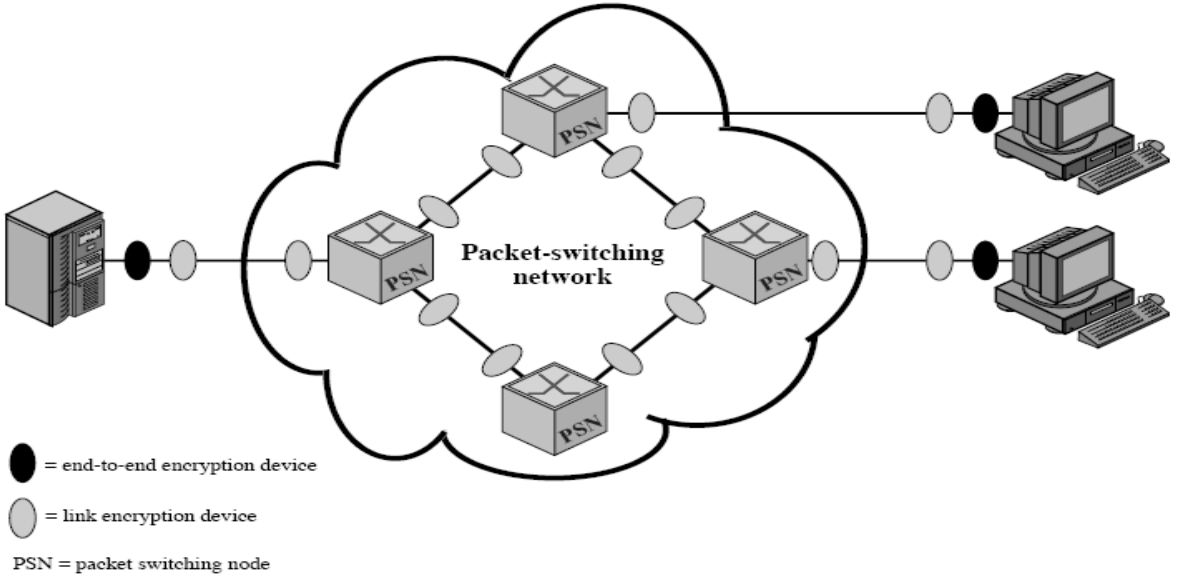
Algorithm	Created By	Key Size	Block Size	Algorithm Structure	Rounds	Cracked?	Existing Cracks
Rijndael	Joan Daemen & Vincent Rijmen in 1998	128 bits, 192 bits, 256 bits	128 Bits	Substitution-Permutation Network	10, 12 or 14	No	Side channel attacks
Twofish	Bruce Schneier in 1993	128 bits, 192 bits or 256 bits	128 bits	Feistel Network	16	No	Truncated differential cryptanalysis
Blowfish	Bruce Schneier in 1993	32-448 bit in steps of 8 bits. 128 bits by default	64 bits	Feistel Network	16	No	Second-order differential attack
RC4	Ron Rivest in 1987	Variable	Variable	Stream	Unknown	Yes	Distinguishers based on weak key schedule
RC2	Ron Rivest in 1987	8-128 bits in steps of 8 bits. 64 bits by default	64 bits	Source-Heavy Feistel Network	16 Mixing 2 Mashing	Yes	Related-Key attack
TripleDES	IBM in 1978	112 bits or 168 bits	64 bits	Feistel Network	48	No	Theoretically possible
DES	IBM in 1975	56 bits	64 bits	Feistel Network	16	Yes	Brute force attack, differential crypanalysis, linear cryptanalysis, Davies' attack

Πηγή: http://www.kellermansoftware.com/t-ArticleStrongestAlgo.aspx , 2008

Τεχνικές κρυπτανάλυσης

- Διαφορική κρυπτανάλυση (differential cryptanalysis)
  - Ξεκινάει με δύο όμοια κείμενα που διαφέρουν μεταξύ τους μόνο για λίγα bits. Χρησιμοποιεί στατιστική ανάλυση των διαφορών που προκύπτουν στα αντίστοιχα κρυπτογραφημένα κείμενα (Biham & Shamir, 1993).
- Γραμμική κρυπτανάλυση (linear cryptanalysis)
  - Χρησιμοποιεί την πόλωση που μπορεί να υπάρχει υπέρ κάποιου συμβόλου αν κάνει κανείς επανειλημμένα XOR μεταξύ επιλεγμένων κομματιών ενός κρυπτοκειμένου
- Κατανάλωση ηλεκτρικής ισχύος
- Ανάλυση χρονισμού

Θέση λειτουργίας συσκευών κρυπτογράφησης



- Κρυπτογράφηση σε κάθε ζεύξη
  - Υψηλή ασφάλεια κατά τη μετάδοση
  - Καμία προστασία όταν τα δεδομένα είναι μέσα στους μεταγωγείς
- Κρυπτογράφηση από άκρο σε άκρο
  - Προστατεύονται τα δεδομένα
  - Όχι όμως το μοτίβο της κίνησης (οι επικεφαλίδες δεν κρυπτογραφούνται)
- Πιθανή η χρήση και των δύο τεχνικών

Τέλος Ενότητας