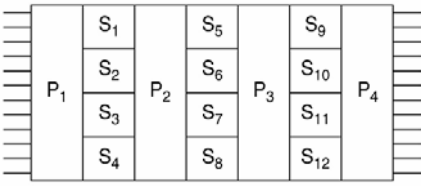
Αλγόριθμοι συμμετρικού κλειδιού

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

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



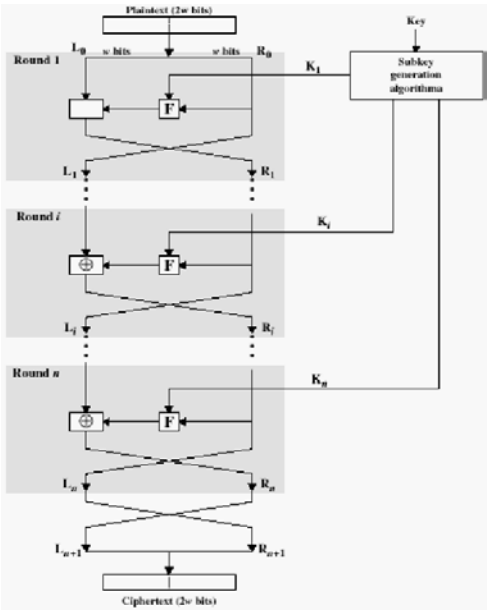
Αν ήταν ένα ενιαίο στάδιο θα είχε 2^{12}=4096 καλώδια

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 που 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

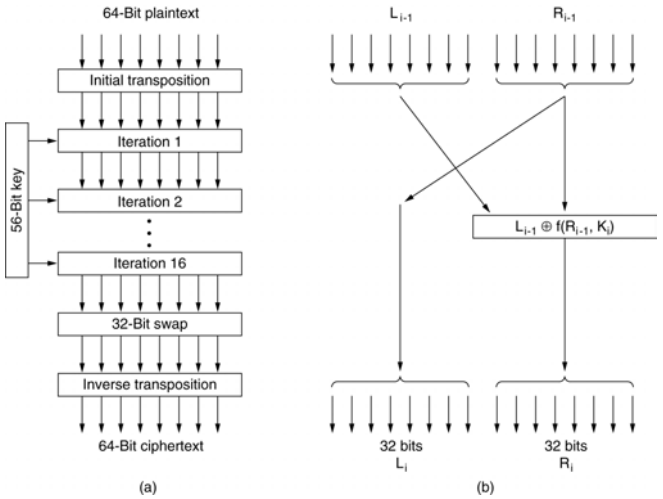


Δρ. Κ. Σ. Χειλάς, Δίκτυα Η/Υ ΙΙΙ, Τ.Ε.Ι. Σερρών, © 2007

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

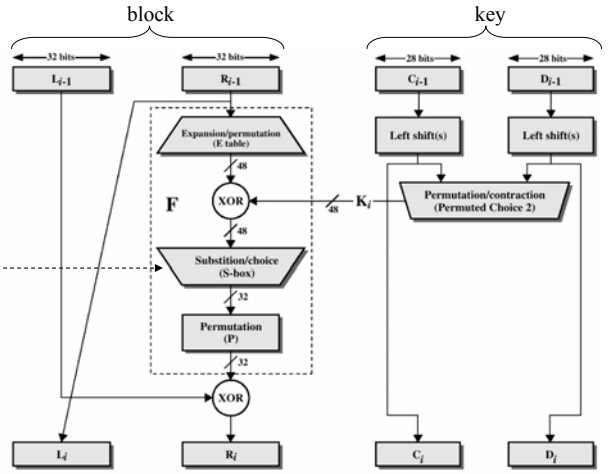
DES



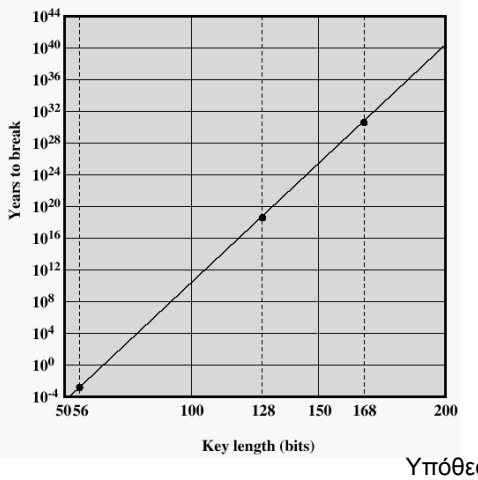
Λεπτομέρεια ενός κύκλου

Λειτουργία ενός κύκλου DES

Τα 48 bits σπάνε σε 8 X 6 bits και περνάνε από 8 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

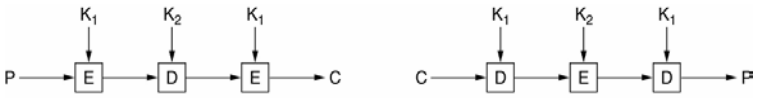


Υπόθεση 10^6 δοκιμές / μsec

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
EK[X] = encryption of X using key K
DK[Y] = decryption of Y using key K

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 (1997) τον αντικαταστάτη του με έναν ανοικτό
- διαγωνισμό κρυπτογραφίας

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)

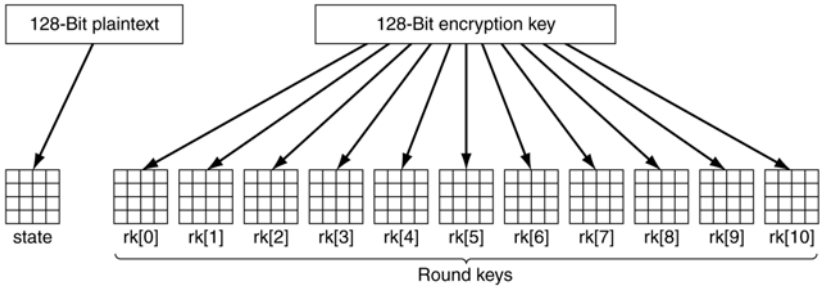
Οκτ. 2000 NIST επέλεξε τον *Rijndael* → 2001: FIPS 197 (Federal Information Processing Standard)

Rijndael

- 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 */
}
```


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.

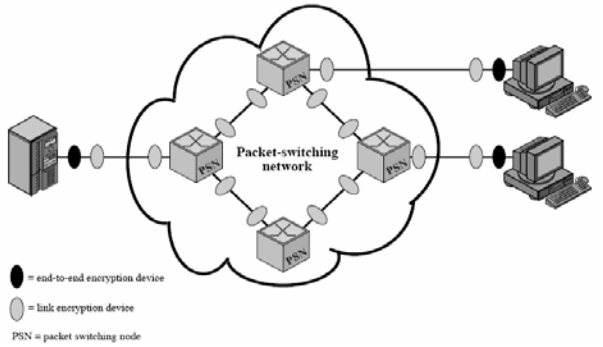
RC5

- 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

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

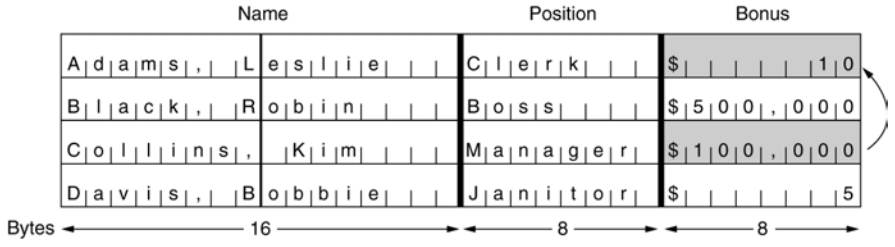
Αλγόριθμοι συμμετρικού κλειδιού

Καταστάσεις/ τρόποι κρυπτογράφησης (Cipher Modes)

Τρόποι κρυπτογράφησης

- Όλοι οι block ciphers είναι στην ουσία μονοαλφαβητικοί κώδικες αντικατάστασης απλά χρησιμοποιούν πιο μεγάλους χαρακτήρες (64 ή 128 Bit)
- Όσες φορές κι αν κωδικοποιήσω το ίδιο κομμάτι κειμένου χρησιμοποιώντας το ίδιο κλειδί, το αποτέλεσμα θα είναι το ίδιο.
- Electronic Code Book (ECB)
- Σε μηνύματα που είναι δομημένα ο κρυπταναλυτής μπορεί να βρει στοιχεία για να αλλοιώσει ή να αντικαταστήσει το περιεχόμενο κάποιων κομματιών.
- Ο στόχος είναι να υπάρχει τρόπος ώστε αν υπάρξουν όμοια κομμάτια κειμένου αυτά να δώσουν διαφορετικά κωδικοποιημένα blocks.

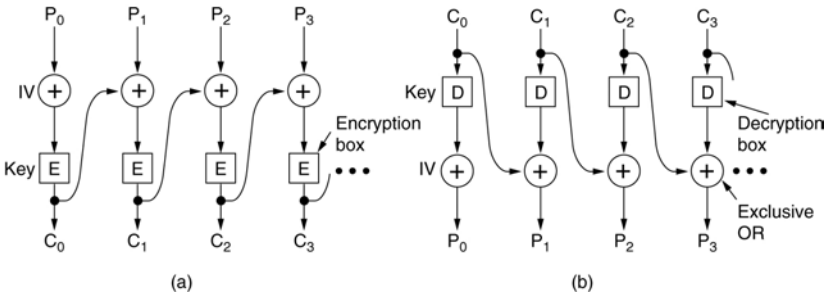
Electronic Code Book Problem



Cipher Block Chaining

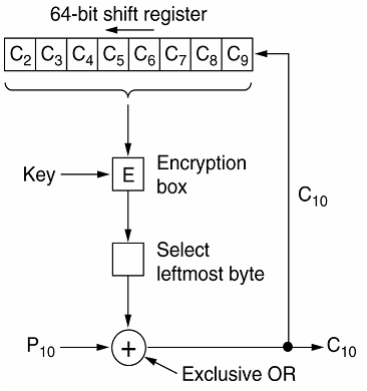
- Για να αντιμετωπιστεί το πρόβλημα ECB
- κάθε block γίνεται XOR με το προηγούμενό του πριν κρυπτογραφηθεί.
- Το προηγούμενο έχει ήδη κρυπτογραφηθεί
- Το πρώτο block γίνεται XOR με ένα τυχαίο διάνυσμα αρχικοποίησης IV (initialization vector) που πρέπει να ανταλλαχθεί μαζί με το κλειδί του κώδικα.

Cipher Block Chaining



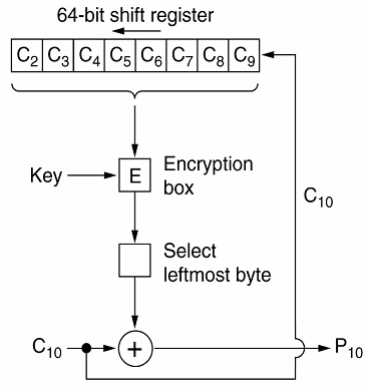
- π.χ. $C_0=E(P_0 \oplus IV)$, $C_1=E(P_1 \oplus IV)$, ….
  $P_0=IV \oplus D(C_0)$
  κύριο πλεονέκτημα είναι ότι λόγω διαφορετικής αρχικοποίησης το ίδιο κείμενο δεν θα δίνει το ίδιο κρυπτογραφημένο, δυσκολεύοντας έτσι τους κρυπταναλυτές

Cipher Feedback Mode κρυπτογράφηση



Τα οκτώ προηγούμενα bits (64 bit) κρυπτογραφούνται και το δεξιότερο Byte του αποτελέσματος γίνεται XOR με το προς αποστολή byte. Αντίγραφο αυτού του byte προωθείται στη δεξιότερη θέση του καταχωρητή ολίσθησης.

Cipher Feedback Mode αποκρυπτογράφηση



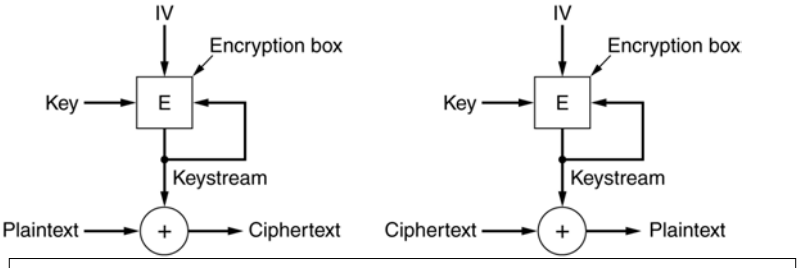
Το κρυπτογραφημένο byte γίνεται XOR με το δεξιότερο Byte από την κρυπτογραφημένη έκδοση των προηγούμενων 8 bytes (64 bit) και δίνει το αρχικό Byte.

Ένα πρόβλημα είναι ότι αν καταστραφεί ένα byte τότε θα υπάρξει σφάλμα στην αποκρυπτογράφηση των 8 bytes στα οποία θα συμμετέχει όσο είναι μέσα στον καταχωρητή. Αυτό όμως είναι ένα μικρό τοπικό πρόβλημα.

Stream Cipher Mode

- Υπάρχουν εφαρμογές για τις οποίες είναι σοβαρό πρόβλημα είναι να καταστρέφονται 64 bit λόγω λάθους σε ένα μόνο bit.
- Σε τέτοιες περιπτώσεις μπορεί να εφαρμοστεί μια άλλη τακτική που λέγεται stream cipher mode.
- Ένα διάνυσμα αρχικοποίησης κωδικοποιείται και δίνει μια νέα κωδικοποιημένη ακολουθία (keystream).
- Σε κάθε βήμα η ακολουθία αυτή κωδικοποιείται εν νέου και το αποτέλεσμα γίνεται XOR με το κείμενο προς κρυπτογράφηση.
- Με άλλα λόγια, το keystream χρησιμοποιείται ως ένα one-time-pad.

Stream Cipher Mode

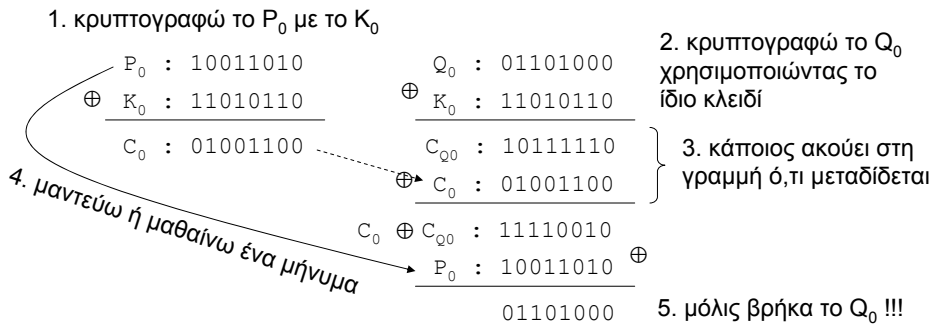


Κρυπτογράφηση σε Stream Mode | Αποκρυπτογράφηση σε Stream Mode

Το ζεύγος {IV, key} δεν πρέπει να χρησιμοποιηθεί ποτέ

Αν συμβεί αυτό ο κώδικας είναι εκτεθειμένος σε "keystream reuse attack"

Παράδειγμα keystream reuse attack



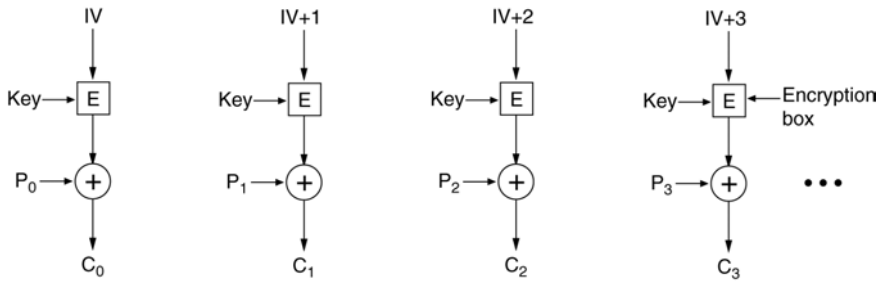
Counter mode

- Το πρόβλημα όλων των παραπάνω τρόπων, εκτός του ECB, είναι ότι για να δω ένα κομμάτι δεδομένων πρέπει να τα αποκρυπτογραφήσω όλα. Αυτό είναι σοβαρό πρόβλημα ειδικά σε περιπτώσεις όπου η πρόσβαση στα δεδομένα γίνεται με μη σειριακό τρόπο, όπως η πρόσβαση σε αρχεία σε έναν σκληρό δίσκο.
- Για να λυθεί αυτό το πρόβλημα εφευρέθηκε μια μέθοδος κρυπτογράφησης που είναι γνωστή με το όνομα counter mode.

Counter mode

- Με τη μέθοδο αυτή το αρχικό κείμενο δεν κρυπτογραφείται άμεσα. Ένα διάνυσμα αρχικοποίησης (IV) κρυπτογραφείται και γίνεται XOR με το πρώτο κομμάτι κειμένου.
- Για κάθε νέο κομμάτι κειμένου το IV αυξάνεται κατά ένα και η διαδικασία επαναλαμβάνεται. Έτσι, κάθε block κειμένου μπορεί να αποκρυπτογραφηθεί ανεξάρτητα από τα υπόλοιπα ανάλογα με τη θέση του.
- Η μέθοδος έχει την ίδια αδυναμία με την προηγούμενη. Είναι ευαίσθητη σε keystream reuse attack.

Counter mode



Δρ. Κ. Σ. Χειλάς, Δίκτυα Η/Υ ΙΙΙ, Τ.Ε.Ι. Σερρών, © 2007

Αλγόριθμοι συμμετρικού κλειδιού

Διανομή κλειδιών

Διανομή κλειδιών

- Η ισχύς κάθε κρυπτογραφικού συστήματος βρίσκεται στη διανομή των κλειδιών.
- Οι δυνατότητες είναι οι εξής:
  1. Ο Α επιλέγει το κλειδί και το δίνει αυτοπροσώπως στον Β
  2. Ένα τρίτο μέρος επιλέγει το κλειδί και το δίνει αυτοπροσώπως στους άλλους δύο
  3. Ο Α και ο Β διαθέτουν ήδη ένα ασφαλές (κρυπτογραφημένο) κανάλι επικοινωνίας και ανταλλάσσουν το κλειδί μέσα από αυτό.
  4. Αν ο Α και ο Β διαθέτουν ήδη ένα ασφαλές κανάλι επικοινωνίας με έναν τρίτο C, ο C μπορεί να παραδώσει το νέο κλειδί στους Α και Β μέσω της σύνδεσης αυτής.

Διανομή κλειδιών

- Προτιμητέα μέθοδος είναι η τέταρτη
- Χρησιμοποιούνται δύο ειδών κλειδιά:
  - Session key: κλειδί μιας χρήσης που χρησιμοποιείται κατά τη διάρκεια μιας επικοινωνίας μεταξύ των μερών.
  - Permanent key: μόνιμο κλειδί γνωστό εκ των προτέρων που δεν ανταλλάσσεται πάνω από το δίκτυο. Χρησιμοποιείται μόνο για την κρυπτογράφηση και αποστολή των session keys.

Στοιχεία υλοποίησης

- Κέντρο διαμοιρασμού κλειδιών (key distribution center): Προσδιορίζει ποιος επιτρέπεται να επικοινωνεί με ποιόν. Όταν δοθεί η εξουσιοδότηση για την επικοινωνία μεταξύ δύο μερών, το KDK παρέχει στους συμμετέχοντες το κλειδί μιας χρήσης (session key)
- Front-end processor: είναι ο επεξεργαστής που εκτελεί την κρυπτογράφηση από άκρο σε άκρο και παραλαμβάνει τα κλειδιά μιας χρήσης εκ μέρους του Η/Υ ή του τερματικού που εκπροσωπεί.

Αυτόματη διανομή κλειδιού