ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

ΚΡΥΠΤΟΓΡΑΦΙΑ

8η Διάλεξη

Κωνσταντίνου Ελισάβετ

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

Άδειες Χρήσης

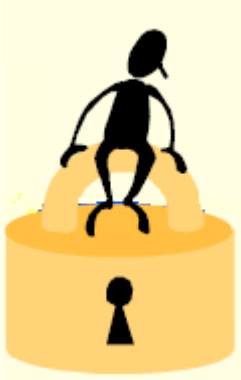
- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αιγαίου**» 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.

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Κρυπτογραφία



Κωνσταντίνου Ελισάβετ
ekonstantinou@aegean.gr

http://www.icsd.aegean.gr/ekonstantinou

AES

Ιαν. 1997: Το NIST (National Institute of Standards and Technology) απευθύνει κάλεσμα για τη δημιουργία νέου προτύπου – Advanced Encryption Standard (AES)

Αυγ. 1998: Πραγματοποιείται το πρώτο συνέδριο για τον AES, παρουσιάζονται 15 υποψήφιοι αλγόριθμοι από 12 χώρες.

Αυγ. 1999: Το NIST ανακοινώνει τους 5 καλύτερους:

MARS (IBM, US)
RC6 (Rivest et al, MIT and RSA, US)
Rijndael (Daemen and Rijmen, Belgium)
Serpent (Anderson, Biham, Knudsen)
Twofish (Schneier, Kelsey et al, Counterpane, US)

Σεπτ. 2000: Επιλέγεται ο Rijndael

Νοεμ. 2001: Νέο πρότυπο NIST FIPS 197

AES

Κριτήρια επιλογής:

Ασφάλεια:

- ✓ Ανθεκτικότητα απέναντι σε γνωστές επιθέσεις
- ✓ Ορθότητα ως προς το μαθηματικό υπόβαθρο
- ✓ Τυχαιότητα
- ✓ Όλα τα παραπάνω σε σχέση με τους άλλους υποψήφιους αλγόριθμους

Κόστος

Υλοποίηση:

- ✓ Ευελιξία (block and key lengths, platforms, H/W vs S/W)
- ✓ Διαφορετικές χρήσεις (modes, stream cipher, hash function, other)
- ✓ Απλότητα

AES

Χρήση 3 μεγεθών κλειδιών: 128, 192 και 256 bits (αντίστοιχα AES-128, AES-192, AES-256).

Όσο αυξάνεται το μέγεθος του κλειδιού τόσο αυξάνεται η ισχύς του αλγορίθμου. Η NSA που υιοθέτησε τον AES για την επικοινωνία των υπηρεσιών της Αμερικής, όρισε το μέγεθος 128 για επικοινωνίες που χαρακτηρίζονται "secret" και το 192 ή 256 μέγεθος κλειδιού για "top secret".

Αν υποθέσουμε ότι έχουμε έναν υπολογιστή που εξετάζει 1 δις κλειδιά το δευτερόλεπτο, τότε για να βρεθεί το κλειδί στον AES με exhaustive search χρειάζονται:

Type	Possible Keys	Time needed in years
AES-128	$2^{128} = 3{,}4028 \cdot 10^{38}$	$1{,}0790 \cdot 10^{22}$
AES-192	$2^{192} = 6{,}2772 \cdot 10^{57}$	$1{,}9904 \cdot 10^{41}$
AES-256	$2^{256} = 1{,}1579 \cdot 10^{77}$	$3{,}6717 \cdot 10^{60}$

AES

AES-128: 9 encryption rounds + 1 final round

AES-192: 11 encryption rounds + 1 final round

AES-256: 13 encryption rounds + 1 final round

Κάθε γύρος περιλαμβάνει τις εξής διαδικασίες: SubBytes, ShiftRows, MixColumns, AddRoundKey.

Ο τελευταίος γύρος δεν περιλαμβάνει τη διαδικασία MixColumns.

Block size = 128 bits

⇒ 4x4 πίνακα όπου κάθε στοιχείο έχει μέγεθος 1 byte.

Επομένως, ο AES λειτουργεί με bytes σε αντίθεση με τον DES που λειτουργούσε με bits. Αυτό κάνει την υλοποίησή του πιο αποδοτική σε software ενώ σε hardware δεν υπάρχουν διαφοροποιήσεις στην ταχύτητα των αλγορίθμων.

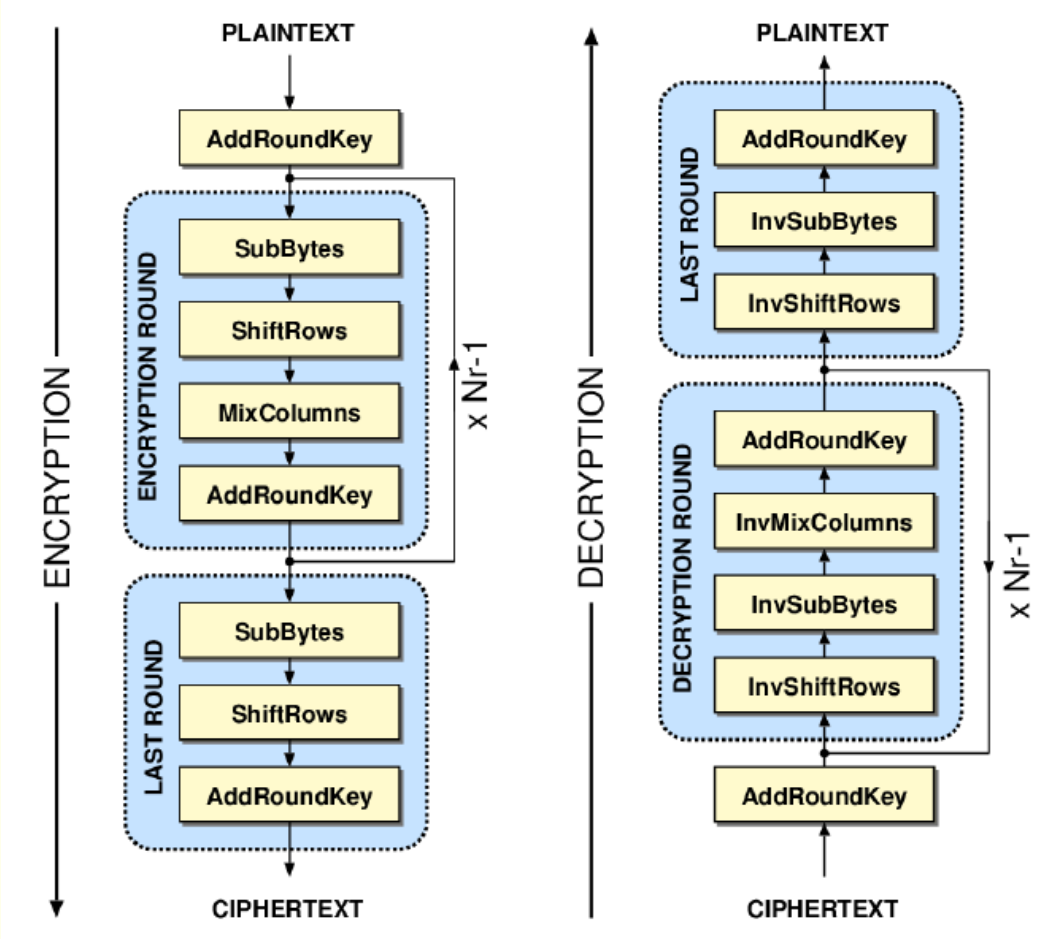
AES

Συνοπτικά, η διαδικασία της κρυπτογράφησης έχει ως εξής:

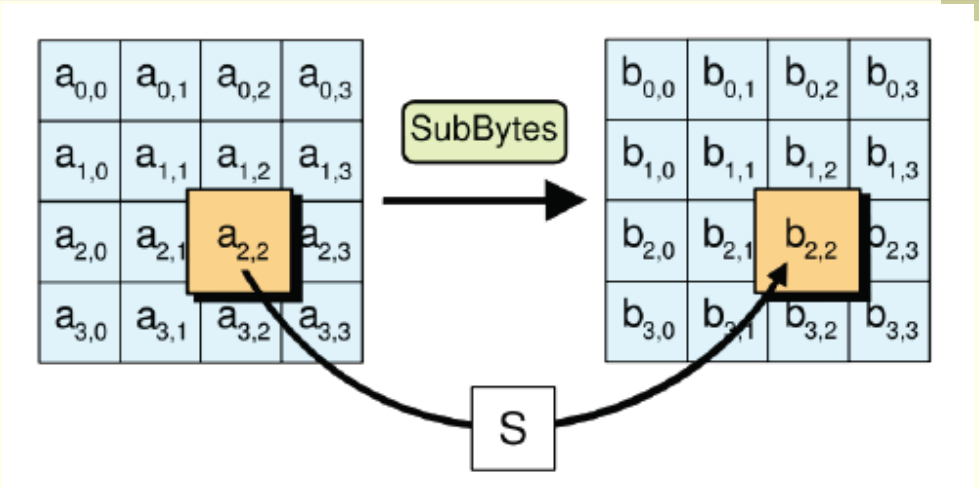
1) Το block των 128 bits οργανώνεται σε έναν 4x4 πίνακα.
2) Εκτελείται η διαδικασία KeyExpansion, όπου δημιουργούνται τα κλειδιά που θα χρησιμοποιηθούν στους επόμενους γύρους.
3) Εκτελείται η διαδικασία AddRoundKey (δηλαδή το αρχικό block γίνεται xor με το 1ο κλειδί).
4) Εκτελούνται οι γύροι του αλγορίθμου (το πλήθος τους Nr εξαρτάται από το μέγεθος του κλειδιού).
5) Τελευταίος γύρος (ίδιος με τους προηγούμενους, μόνο που τώρα δεν υπάρχει η διαδικασία MixColumns).

Η αποκρυπτογράφηση γίνεται με τις αντίστροφες διαδικασίες.

AES


ENCRYPTION
PLAINTEXT
AddRoundKey
ENCRYPTION ROUND
SubBytes
ShiftRows
MixColumns
AddRoundKey
x Nr-1
LAST ROUND
SubBytes
ShiftRows
AddRoundKey
CIPHERTEXT
DECRYPTION
PLAINTEXT
LAST ROUND
AddRoundKey
InvSubBytes
InvShiftRows
DECRYPTION ROUND
AddRoundKey
InvMixColumns
InvSubBytes
InvShiftRows
x Nr-1
AddRoundKey
CIPHERTEXT

Διαδικασία SubBytes



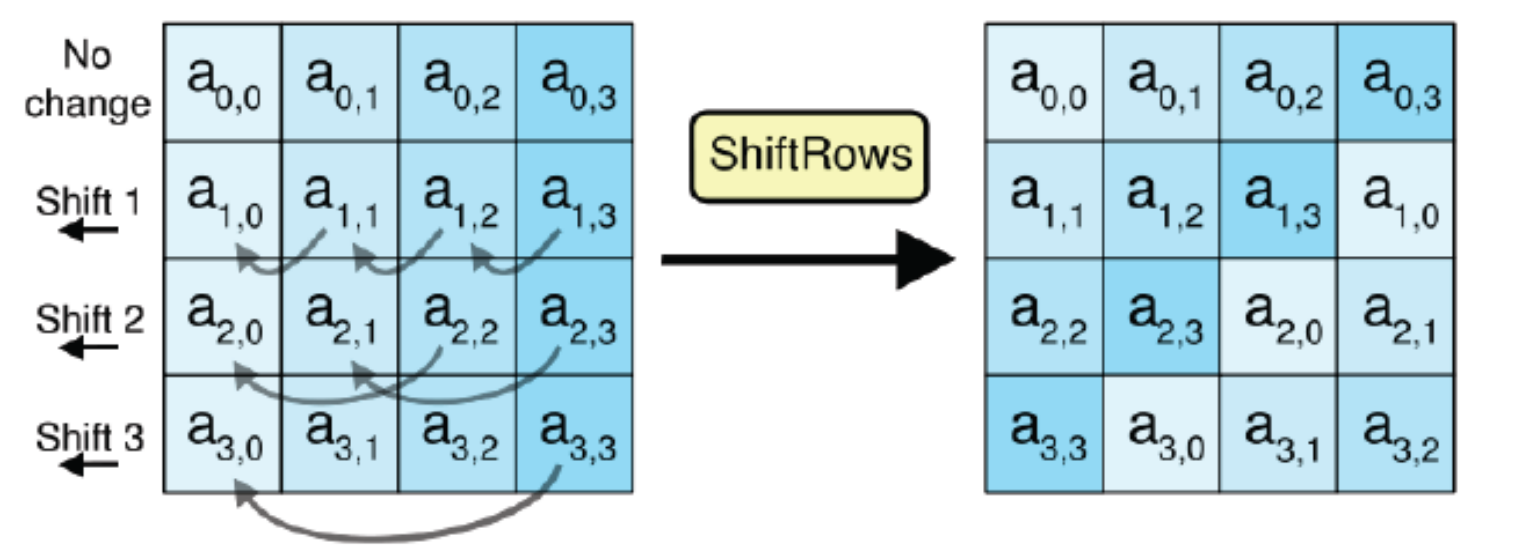
$$\begin{bmatrix} b7 \\ b6 \\ b5 \\ b4 \\ b3 \\ b2 \\ b1 \\ b0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} a7 \\ a6 \\ a5 \\ a4 \\ a3 \\ a2 \\ a1 \\ a0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Rijndaels S-Box

$$\begin{bmatrix} a7 \\ a6 \\ a5 \\ a4 \\ a3 \\ a2 \\ a1 \\ a0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} b7 \\ b6 \\ b5 \\ b4 \\ b3 \\ b2 \\ b1 \\ b0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

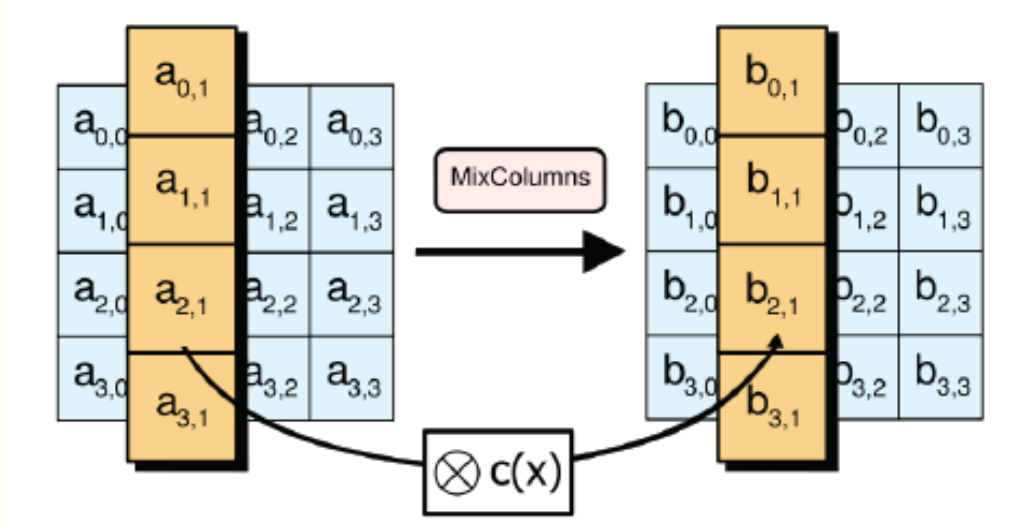
Rijndaels inverse S-Box

Διαδικασία ShiftRows



Η αντίστροφη διαδικασία InvShiftRows είναι ακριβώς η ίδια, μόνο που η μετατόπιση γίνεται τώρα προς τα δεξιά.

Διαδικασία MixColumns



Ουσιαστικά, αν κάθε στήλη γραφεί ως πολυώνυμο 3ου βαθμού στο GF(2^8), τότε η διαδικασία MixColumns είναι ένας πολ/σμός με το πολυώνυμο c(x) ο οποίος ακολουθείται από μια αναγωγή με το x^4+1.

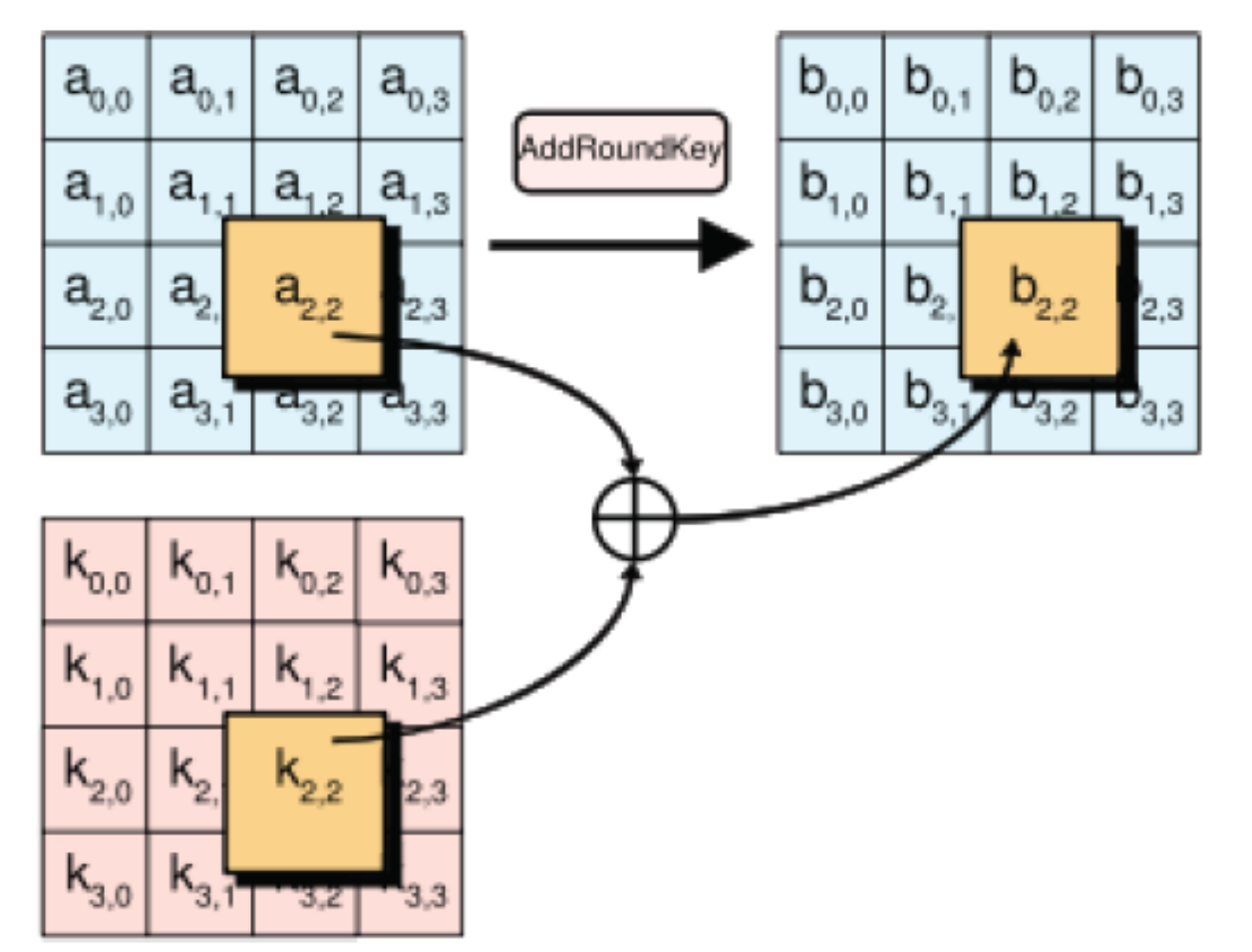
Η διαδικασία InvMixColumns είναι ακριβώς ίδια, μόνο που τώρα γίνεται πολ/σμός με το πολυώνυμο $c^{-1}(x)$.

Διαδικασία MixColumns

Αν υποθέσουμε ότι a_0, a_1, a_2 και a_3 είναι τα bytes κάποιας στήλης, τότε αυτή μετασχηματίζεται στη στήλη b_0, b_1, b_2 και b_3 ως εξής:

Multiplication				Hexadecimal				Mathematical
2	3	1	1	02	03	01	01	$b_0 = 2a_0 + a_3 + a_2 + 3a_1$
1	2	3	1	01	02	03	01	$b_1 = 2a_1 + a_0 + a_3 + 3a_2$
1	1	2	3	01	01	02	03	$b_2 = 2a_2 + a_1 + a_0 + 3a_3$
3	1	1	2	03	01	01	02	$b_3 = 2a_3 + a_2 + a_1 + 3a_0$

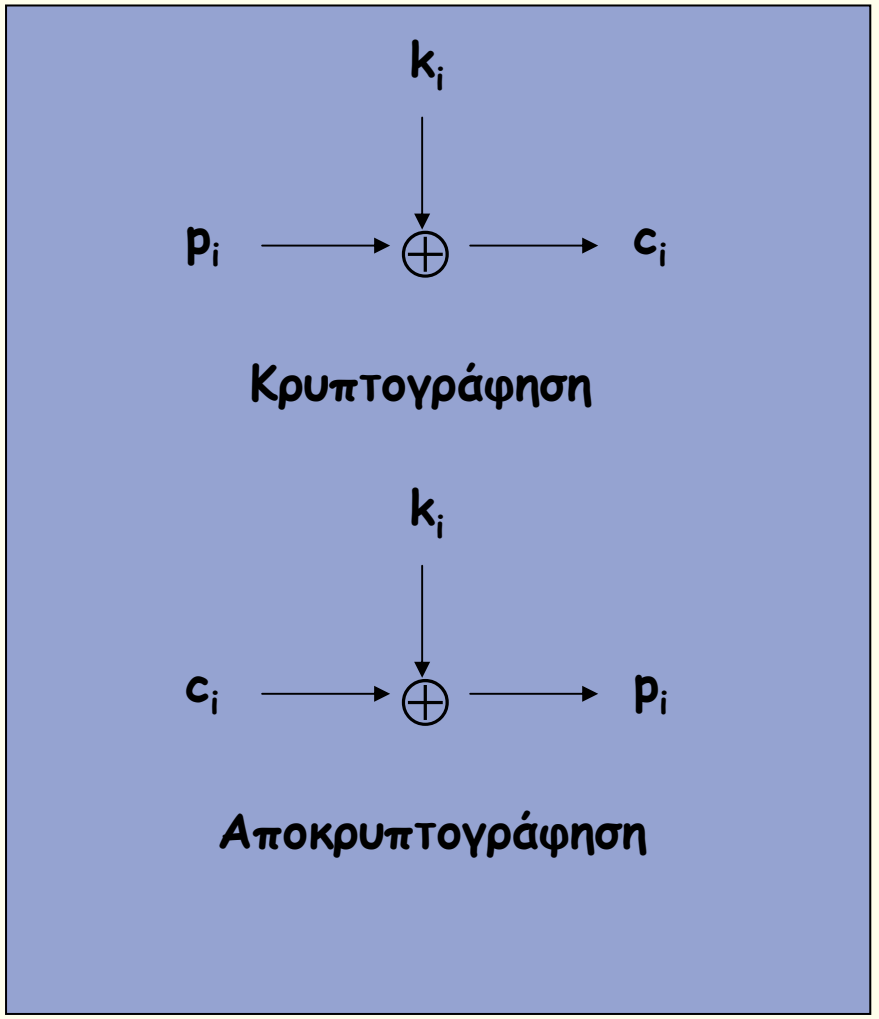
Διαδικασία AddRoundKey



Stream ciphers

Η διαδικασία κωδικοποίησης για έναν stream cipher συνοψίζεται παρακάτω:

1. Το αρχικό μήνυμα (plaintext) μετατρέπεται σε δυαδική ακολουθία.
2. Επιλέγεται ένα κλειδί κρυπτογράφησης το οποίο μετατρέπεται σε δυαδική ακολουθία επίσης.
3. Το αρχικό μήνυμα και το κλειδί προστίθενται σύμφωνα με την πράξη XOR για να προκύψει το ciphertext.



Stream Ciphers

Γενικά είναι γρηγορότεροι από τους block ciphers σε υλοποιήσεις σε υλικό. Μπορεί να είναι συμμετρικοί ή δημοσίου κλειδιού ανάλογα με τον τρόπο που προκύπτει το keystream.

Ένας synchronous stream cipher έχει keystream που δημιουργείται ανεξάρτητα από το plaintext και το ciphertext (π.χ. OFB ή LFSR).

Ένας self-synchronizing ή asynchronous stream cipher έχει keystream που δημιουργείται από μια συνάρτηση του κλειδιού και κάποιων bits του ciphertext (π.χ. CFB).

LFSR – Linear Feedback Shift Registers:

- Γρήγορη υλοποίηση σε hardware
- Παράγουν σειρές με μεγάλη περίοδο
- Παράγουν σειρές με καλές στατιστικές ιδιότητες
- Μπορούν να αναλυθούν εύκολα με αλγεβρικές τεχνικές

Linear Feedback Shift Registers

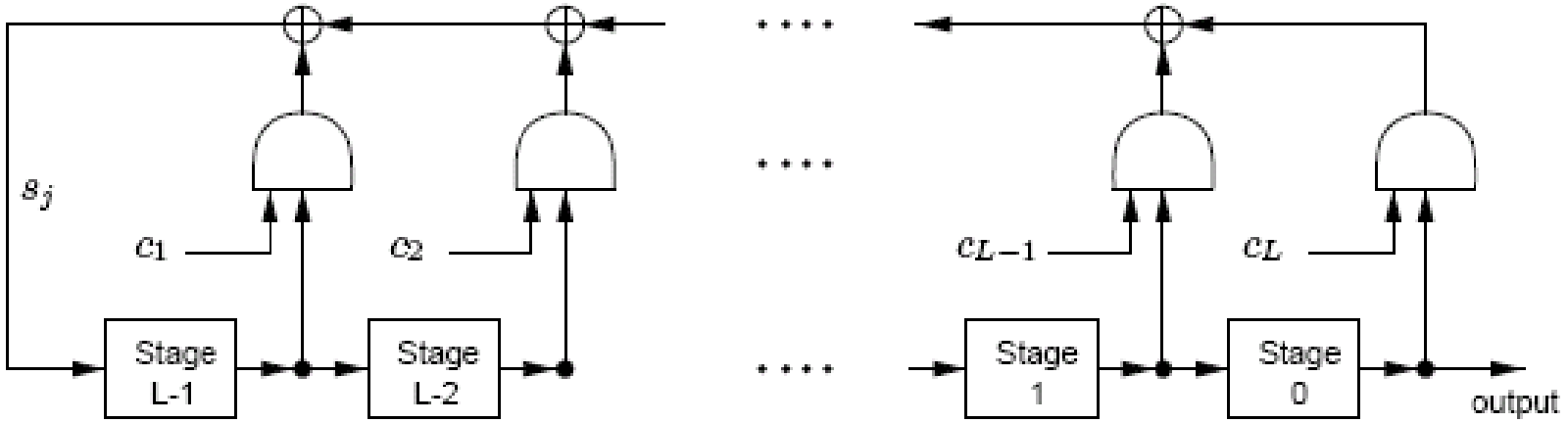


Figure 6.4: *A linear feedback shift register (LFSR) of length L.*

$$s_j = (c_1 s_{j-1} + c_2 s_{j-2} + \cdots + c_L s_{j-L}) \bmod 2 \quad \text{for } j \geq L.$$

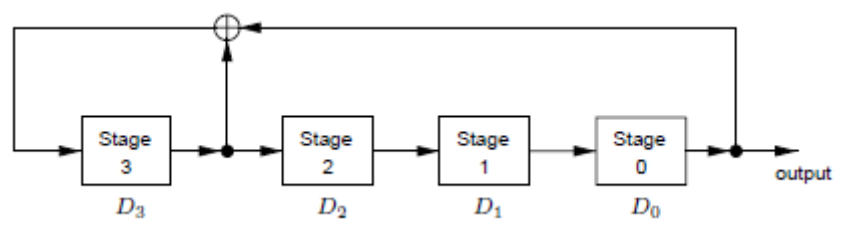
Linear Feedback Shift Registers

Ο LFSR του σχήματος συμβολίζεται με <L, C(D)> όπου

$C(D) = 1 + c_1D + c_2D^2 + \ldots + c_LD^L$ είναι το connection polynomial.

Αν ο βαθμός του πολυωνύμου είναι L, δηλαδή $c_L = 1$, τότε ο LFSR καλείται non-singular. Οι τιμές $[s_{L-1}, s_{L-2}, \ldots, s_0]$ αποτελούν το initial state του LFSR.

Στο παρακάτω σχήμα ποιο το connection polynomial?



Τι συμβαίνει αν $c_i = 0$ και τι αν $c_i = 1$?

Linear Feedback Shift Registers

Έστω C(D) το connection polynomial ενός LFSR.

1) Αν το C(D) είναι irreducible (ανάγωγο) πολυώνυμο στο Z_2, τότε κάθε μία από τις $2^L - 1$ μη μηδενικές αρχικές καταστάσεις δημιουργεί έξοδο με περίοδο ίση με τον μικρότερο θετικό ακέραιο N τέτοιο ώστε το C(D) να διαιρεί το πολυώνυμο $1 + D^N$ στο $Z_2[D]$ (το N πάντα σε αυτή την περίπτωση είναι διαιρέτης του $2^L - 1$).
2) Αν το C(D) είναι primitive (πρωταρχικό) πολυώνυμο, τότε κάθε μία από τις $2^L - 1$ μη μηδενικές αρχικές καταστάσεις ενός non-singular LFSR δημιουργεί έξοδο με τη μέγιστη δυνατή περίοδο, δηλαδή ίση με $2^L - 1$.

Ανάγωγο πολυώνυμο: δεν διαιρείται με κανένα πολυώνυμο στο Z_2

Πρωταρχικό πολυώνυμο: ένα ανάγωγο πολυώνυμο f(x) βαθμού m καλείται πρωταρχικό αν διαιρεί το πολυώνυμο $x^k - 1$ για $k = 2^m - 1$.

Linear Feedback Shift Registers

Αν C(D) είναι πρωταρχικό πολυώνυμο με βαθμό L, τότε ο <L, C(D)> LFSR καλείται maximum-length LFSR. Η έξοδος που προκύπτει από ένα maximum-length LFSR με μη-μηδενική είσοδο καλείται m-sequence.

Στατιστικές ιδιότητες των m-sequences:

Έστω s μια m-sequence που δημιουργείται από ένα maximum-length LFSR μήκους L.

(1) Έστω k ένας θετικός ακέραιος με $k < L+1$ και έστω s^* μια υποακολουθία του s μήκους $2^L + k - 2$. Τότε κάθε μη-μηδενική ακολουθία μήκους k εμφανίζεται 2^{L-k} φορές ως μια υποακολουθία του s^*. Η μηδενική ακολουθία εμφανίζεται $2^{L-k}-1$ φορές. Δηλαδή η κατανομή των ακολουθιών με μήκος το πολύ L είναι σχεδόν ομοιόμορφη.

(2) Η ακολουθία s ικανοποιεί τα κριτήρια του Golomb. Δηλαδή, κάθε m-sequence είναι επίσης μια pn-sequence.

Τα κριτήρια του Golomb

Run καλείται μια υποακολουθία της s που αποτελείται είτε από διαδοχικά 0 ή διαδοχικά 1.

Η συνάρτηση αυτοσυσχέτισης της s είναι ίση με:

$$C(t) = \frac{1}{N} \sum_{i=0}^{N-1} (2s_i - 1) \cdot (2s_{i+t} - 1), \quad \text{for } 0 \leq t \leq N-1.$$

Η συνάρτηση αυτοσυσχέτισης μετράει το πόσο όμοια είναι η s με μία μετατόπιση της κατά t θέσεις. Αν η s είναι πραγματικά τυχαία περιοδική ακολουθία με περίοδο N, τότε η ποσότητα |N*C(t)| θα είναι αρκετά μικρή για όλες τις τιμές του t.

Τα κριτήρια του Golomb

Έστω s μια ακολουθία με περίοδο N.

1) Στον κύκλο $s = s_0 s_1 s_2 \ldots s_N$ ο αριθμός των 1 και των 0 πρέπει να διαφέρουν το πολύ κατά 1.
2) Στον κύκλο s, τουλάχιστον τα μισά runs θα πρέπει να έχουν μήκος 1, τουλάχιστον το $\frac{1}{4}$ των runs θα πρέπει να έχουν μήκος 2, τουλάχιστον το 1/8 των runs να έχει μήκος 3 κ.ο.κ.
3) Η συνάρτηση αυτοσυσχέτισης θα πρέπει να επιστρέφει μόνο 2 τιμές.

Π.χ. η s = 011001000111101 με N = 15 ικανοποιεί τα 3 κριτήρια του Golomb.

Κάθε ακολουθία που ικανοποιεί αυτά τα κριτήρια καλείται pn-ακολουθία.

Linear Complexity

Ορισμός: Η γραμμική πολυπλοκότητα μια άπειρα μεγάλης δυαδικής ακολουθίας s θα συμβολίζεται με L(s) και ορίζεται ως εξής:

(i) Αν s είναι η μηδενική ακολουθία s=0000…. τότε L(s) = 0.

(ii) Αν κανένα LFSR δεν μπορεί να «γεννήσει» την ακολουθία s, τότε L(s) = ∞.

(iii) Σε διαφορετική περίπτωση, η γραμμική πολυπλοκότητα L(s) είναι ίση με το μήκος του συντομότερου (shortest) LFSR που δημιουργεί το s.

Ορισμός: Η γραμμική πολυπλοκότητα μιας πεπερασμένου μήκους δυαδικής ακολουθίας s^n θα συμβολίζεται με $L(s^n)$ και είναι ίση με το μήκος του συντομότερου LFSR που στα πρώτα n bits της εξόδου του παράγει την ακολουθία s^n.

Linear Complexity

Αν το πολυώνυμο C(D) είναι ανάγωγο στο Z_2 και έχει βαθμό L, τότε κάθε μία από τις $2^L - 1$ μη-μηδενικές αρχικές καταστάσεις παράγει μια ακολουθία με γραμμική πολυπλοκότητα ίση με L.

Αν $s=s_0, s_1, \ldots$ είναι μια δυαδική ακολουθία και L_N είναι η γραμμική πολυπλοκότητα του s^N, τότε η ακολουθία $L_1, L_2, \ldots$ καλείται linear complexity profile της s.

Βάζοντας τα σημεία (N, L_N) μιας τυχαίας ακολουθίας s στο διδιάστατο χώρο, θα πρέπει να ακολουθούν τη γραμμή L=N/2. Ωστόσο, υπάρχουν ακολουθίες που ικανοποιούν αυτή τη συνθήκη, αλλά δεν είναι τυχαίες:

$$s_i = \begin{cases} 1, & \text{if } i = 2^j - 1 \text{ for some } j \geq 0, \\ 0, & \text{otherwise,} \end{cases}$$

(εδώ $L_N = \left\lfloor \frac{N+1}{2} \right\rfloor$)

Nonlinear FSRs

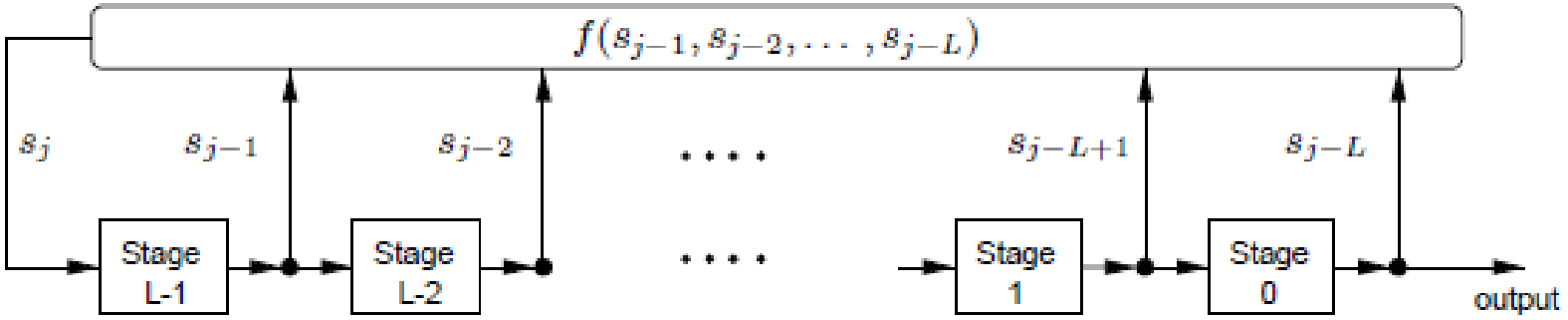


Figure 6.7: *A feedback shift register (FSR) of length L.*

$$s_j = f(s_{j-1}, s_{j-2}, \ldots, s_{j-L}) \text{ for } j \geq L.$$

Nonlinear FSRs

Ένας FSR καλείται non-singular αν και μόνο αν κάθε έξοδός του είναι μια περιοδική ακολουθία. Επιπλέον, κάθε FSR είναι non-singular αν και μόνο αν η συνάρτηση f είναι ίση με $f = s_{j-L} \oplus g(s_{j-1}, s_{j-2}, \cdots, s_{j-L+1})$

Η περίοδος των ακολουθιών που προκύπτουν από ένα non-singular FSR μήκους L είναι το πολύ 2^L. Αν η περίοδος είναι ακριβώς 2^L, τότε ο FSR καλείται de Bruijn FSR και οι ακολουθίες που προκύπτουν από έναν τέτοιο FSR καλούνται ακολουθίες de Bruijn.

Στατιστικές ιδιότητες:

Έστω s μια ακολουθία de Bruijn που δημιουργήθηκε από έναν de Bruijn FSR μήκους L. Έστω k ένας θετικός ακέραιος με $k < L+1$ και έστω s^* μια υποακολουθία του s μήκους $2^L + k - 1$. Τότε κάθε ακολουθία μήκους k εμφανίζεται 2^{L-k} φορές ως μια υποακολουθία του s^*. Δηλαδή η κατανομή των ακολουθιών με μήκος το πολύ L είναι σχεδόν ομοιόμορφη.

Stream Ciphers βασισμένοι σε LFSRs

Παρόλα τα πλεονεκτήματα των LFSRs (αποδοτική υλοποίηση, μεγάλη περίοδο, καλές στατιστικές ιδιότητες), η έξοδός τους μπορεί να προβλεφθεί αρκετά εύκολα.

⇒ Υποθέστε ότι η έξοδος s ενός LFSR έχει γραμμική πολυπλοκότητα L. Το πολυώνυμο C(D) μπορεί να υπολογιστεί εύκολα από τον αλγόριθμο των Berlekamp-Massey γνωρίζοντας κανείς μια υποακολουθία t του s με μήκος τουλάχιστον 2L. Αφού έχει βρεθεί το πολυώνυμο C(D), τότε ο LFSR μπορεί να αρχικοποιηθεί με οποιαδήποτε L bits του t και να παράγει την ακολουθία s.

Πως μπορείς να βρεις την υποακολουθία t?

⇒ Known or chosen plaintext attacks

Ποιο είναι το μυστικό κλειδί σε αυτούς τους stream ciphers?

⇒ Η αρχική κατάσταση και (πιθανώς) το πολυώνυμο C(D)

Stream Ciphers βασισμένοι σε LFSRs

Για να υπάρξει ασφάλεια έναντι τέτοιων επιθέσεων, ένας LFSR δεν πρέπει να χρησιμοποιείται ποτέ ως keystream generator.

Αντιμετώπιση προβλήματος:

1) Μη γραμμικός συνδυασμός των εξόδων πολλών LFSRs
2) Μη γραμμική συνάρτηση των περιεχομένων ενός LFSR
3) Χρήση ενός ή περισσοτέρων LFSRs που ελέγχουν την έξοδο ενός ή περισσοτέρων LFSRs

Επιθυμητές ιδιότητες των keystream generators που βασίζονται σε LFSRs:

1) Μεγάλη περίοδος των ακολουθιών (χρήση maximum-length LFSRs)
2) Μεγάλη γραμμική πολυπλοκότητα
3) Καλές στατιστικές ιδιότητες (π.χ. m-sequences)

Nonlinear combination generators

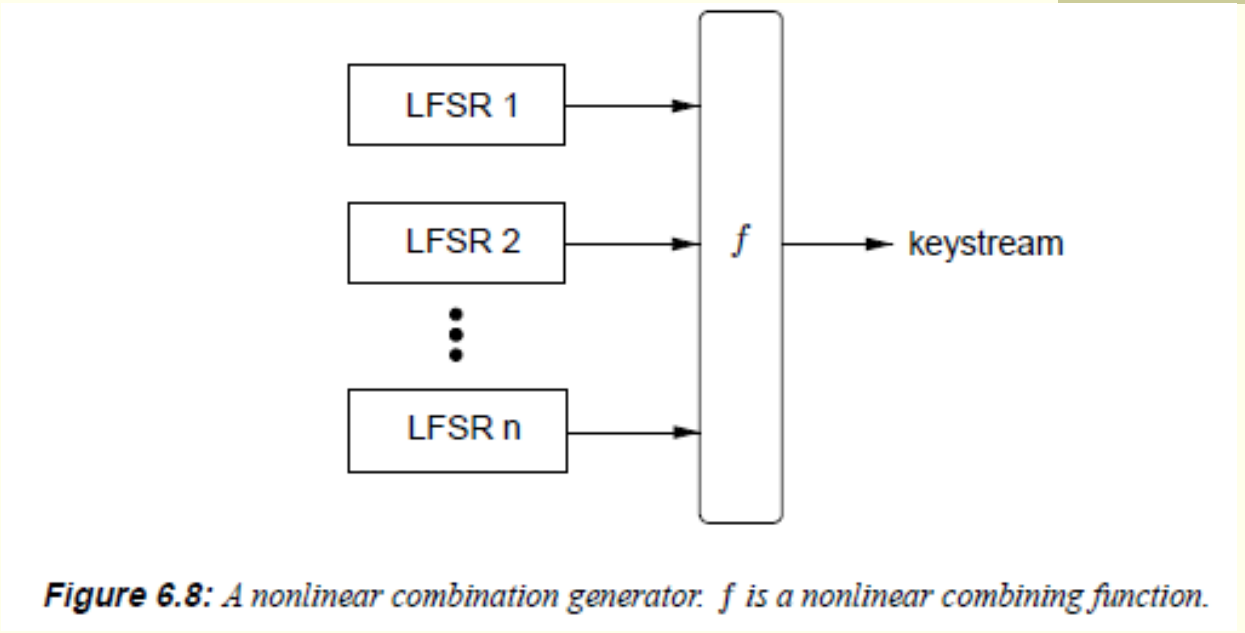


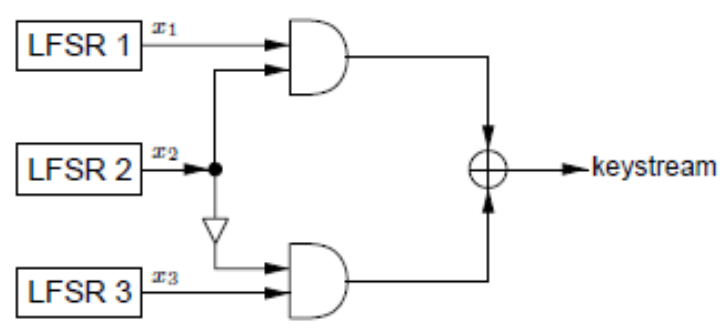
Figure 6.8: *A nonlinear combination generator. f is a nonlinear combining function.*

Π.χ. $f(x_1, x_2, x_3, x_4, x_5) = 1 \oplus x_2 \oplus x_3 \oplus x_4x_5 \oplus x_1x_3x_4x_5$

Nonlinear combination generators

Υποθέστε ότι n maximum-length LFSRs με μήκος αντίστοιχα $L_1, L_2, \ldots, L_n$ συνδέονται μέσω μιας μη-γραμμικής συνάρτησης f(). Τότε η γραμμική πολυπλοκότητα της μη-γραμμικής γεννήτριας που προκύπτει είναι ίση με $f(L_1, L_2, \ldots, L_n)$.

Π.χ. Geffe generator, $f(x_1, x_2, x_3) = x_1x_2 \oplus (1 + x_2)x_3 = x_1x_2 \oplus x_2x_3 \oplus x_3.$

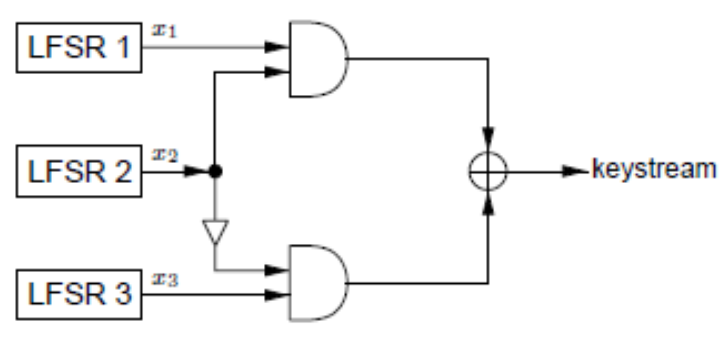


Περίοδος: $(2^{L_1} - 1) \cdot (2^{L_2} - 1) \cdot (2^{L_3} - 1)$

Γραμμική πολυπλοκότητα: $L = L_1L_2 + L_2L_3 + L_3$

Geffe generator

Η γεννήτρια δεν είναι κρυπτογραφικά ασφαλής γιατί πληροφορία για την έξοδο του 1ου και 3ου LFSR προκύπτει πολύ εύκολα από την τελική έξοδο.

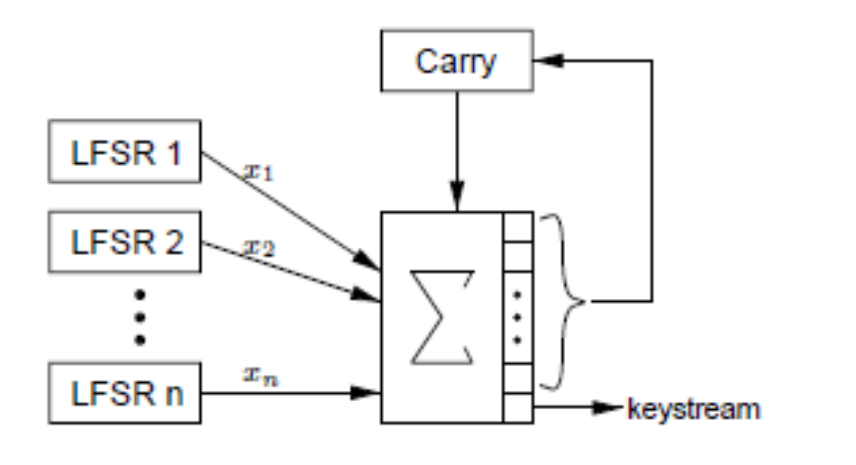


$$P(z(t) = x_1(t)) = P(x_2(t) = 1) + P(x_2(t) = 0) \cdot P(x_3(t) = x_1(t)) = \frac{3}{4}$$

$$P(z(t) = x_3(t)) = \frac{3}{4}$$

Correlation attacks: ενώ σε μια ασφαλή γεννήτρια χρειάζονται $\prod_{i=1}^{n}(2^{L_i} - 1)$ trails, αν η γεννήτρια είναι ευπαθής σε αυτές τις επιθέσεις απαιτούνται $\sum_{i=1}^{n}(2^{L_i} - 1)$

Summation generator



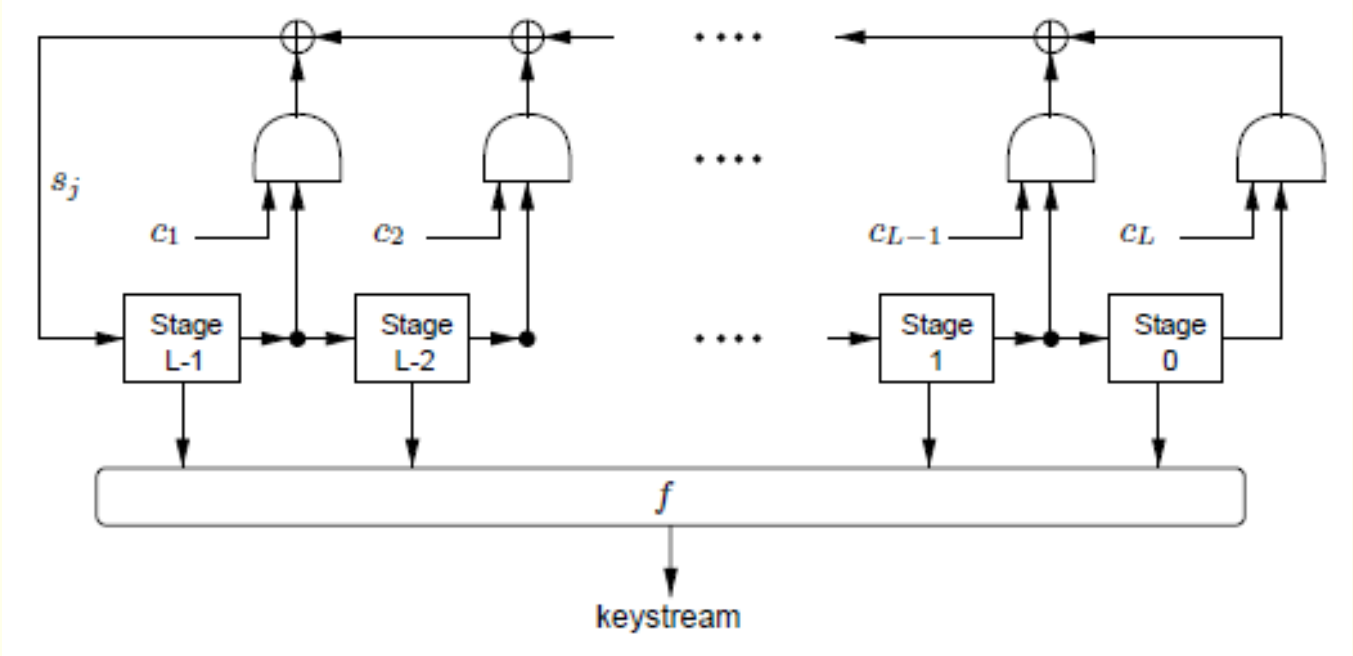
$$S_j = \sum_{i=1}^{n} x_i + C_{j-1}$$

Keystream = S_j mod 2

$C_j = \left\lfloor \frac{S_j}{2} \right\rfloor$

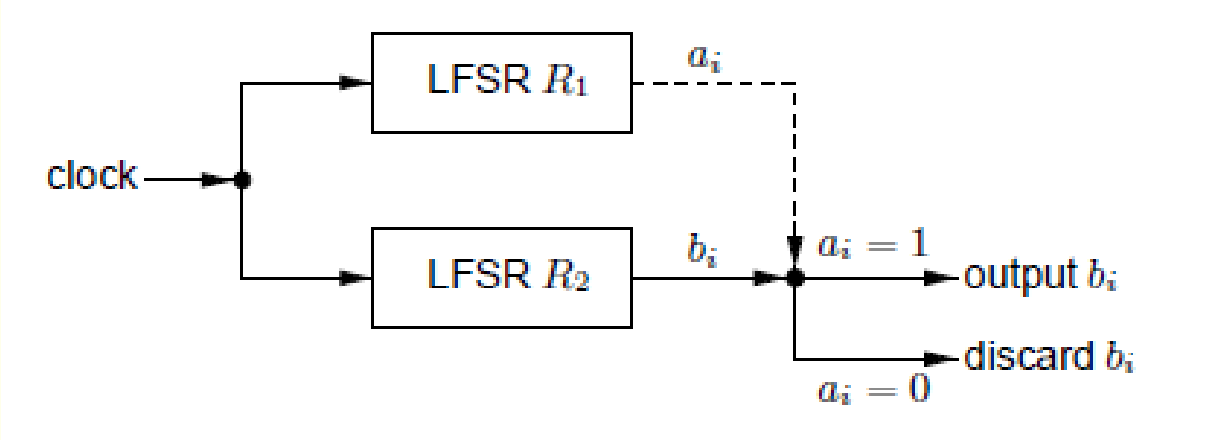
Περίοδος = Γραμμική πολυπλοκότητα = $\prod_{i=1}^{n} (2^{L_i} - 1)$

Nonlinear filter generators



Clock-controlled generators

The shrinking generator:



Αν gcd(L_1, L_2) = 1, τότε η περίοδος των ακολουθιών είναι $(2^{L_2} - 1) \cdot 2^{L_1 - 1}$

Η γραμμική πολυπλοκότητα ικανοποιεί τη σχέση:

$$L_2 \cdot 2^{L_1 - 2} < L(x) \leq L_2 \cdot 2^{L_1 - 1}$$

Η αποδοτικότερη επίθεση απαιτεί $O(2^{L_1} \cdot L_2^3)$ βήματα.

Διάβασμα…

Κεφάλαια 6.1, 6.2 και 6.3 του
Handbook of Applied Cryptography