Ασφάλεια Υπολογιστικών Συστημάτων

7ο Εξάμηνο

Συναρτήσεις Κατακερματισμού και Πιστοποίηση Μηνύματος

Διδάσκων : Δρ. Παρασκευάς Κίτσος, Καθηγητής

https://ecsalab.ece.uop.gr/

Εργαστήριο Ηλεκτρονικών Κυκλωμάτων, Συστημάτων και Εφαρμογών (ECSA Lab.)

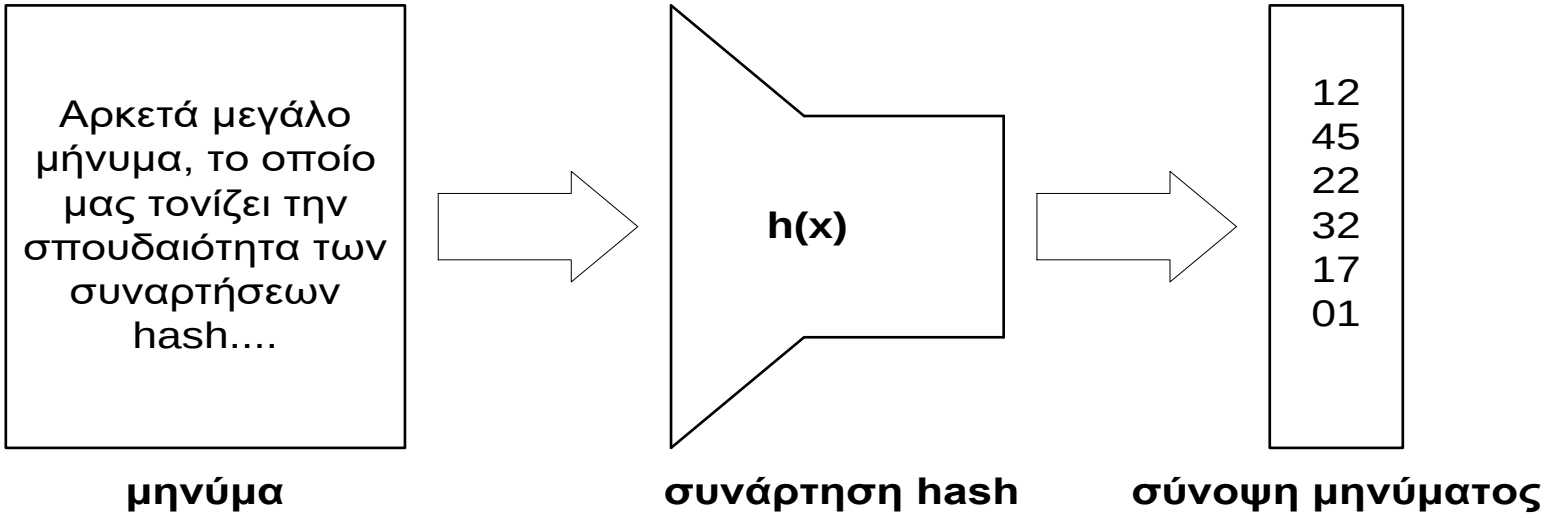
e-mail: kitsos@uop.gr

ΔΙΑΡΘΡΩΣΗ ΕΝΟΤΗΤΑΣ

- Hash functions (Συναρτήσεις κατακερματισμού)
- Μεθοδολογίες Πιστοποίησης Μηνύματος
- Ψηφιακή υπογραφή
- Hash Message Authentication Code (HMAC)

HASH ΣΥΝΑΡΤΗΣΕΙΣ (1/3)

- Μια συνάρτηση hash h: $F^* \rightarrow G^n$ αντιστοιχεί ένα στοιχείο $x \in F^*$ αυθαίρετου μήκους, στο στοιχείο $y \in G^n$ με συγκεκριμένο μήκος n, y=h(x).
- Η συνάρτηση hash δέχεται ένα συγκριτικά μεγάλο μήνυμα και παράγει μια ***σύνοψη*** του μηνύματος μικρότερου και σταθερού μεγέθους.
- Το μέγεθος της σύνοψης αναμένεται να είναι πάνω από 160-bit.



HASH ΣΥΝΑΡΤΗΣΕΙΣ (2/3)

- Η κρυπτογραφική συνάρτηση hash έχει τις παρακάτω ιδιότητες.

  1) δοθέντος ενός y, είναι υπολογιστικά αδύνατο να βρεθεί x τέτοιο ώστε $h(x)=y$.

  2) δοθέντων x, $h(x)$, είναι υπολογιστικά αδύνατο να βρεθεί x' τέτοιο ώστε $h(x)=h(x')$.

  3) είναι υπολογιστικά αδύνατο να βρεθούν x_1, $x_2 \in F^*$, τέτοια ώστε $h(x_1)=h(x_2)$.
- Οι συναρτήσεις που διατηρούν τις (1) και (2) ονομάζονται ***μονόδρομες hash συναρτήσεις (one-way hash functions)***
- Οι συναρτήσεις που διατηρούν τις (2) και (3) ονομάζονται ***ανθεκτικές σε συγκρούσεις hash συναρτήσεις (collision resistance hash functions)***
- Η ιδιότητα (2) ονομάζεται ***ασθενής αντίσταση σε συγκρούσεις***
- Η ιδιότητα (3) ονομάζεται ***ισχυρή αντίσταση σε συγκρούσεις***

HASH ΣΥΝΑΡΤΗΣΕΙΣ (3/3)

- Μονόδρομες hash συναρτήσεις
  - Συναρτήσεις άνευ κλειδιού
  - Συναρτήσεις με κλειδί

ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗ

- Ένα σύστημα θα πρέπει να έχει τη δυνατότητα να πιστοποιεί ότι:

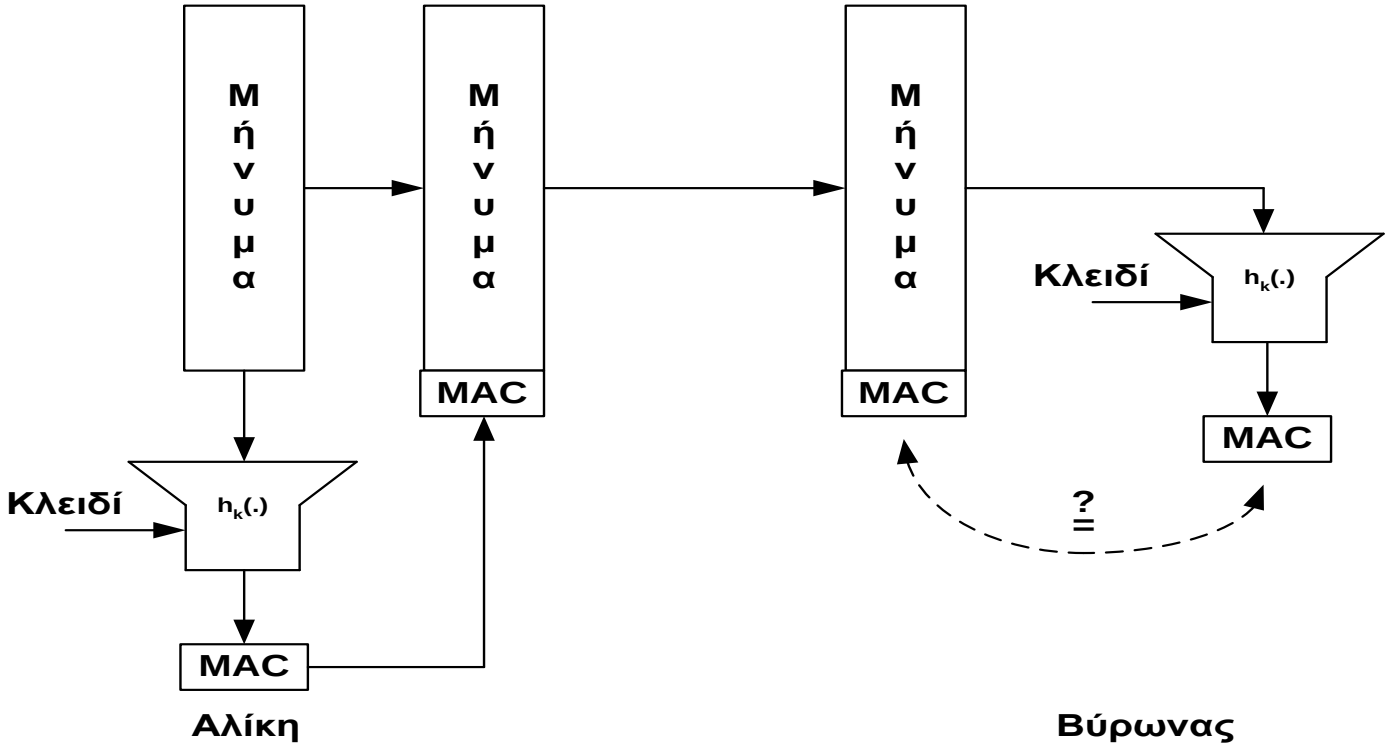
  1. *Ένα μήνυμα έχει αποσταλεί από συγκεκριμένη πηγή ή αποστολέα*
  2. *Τα περιεχόμενα του μηνύματος δεν έχουν διαφοροποιηθεί*
  3. *Το δεδομένα έχουν αποσταλεί σε συγκεκριμένη χρονική περίοδο και με συγκεκριμένη διαδοχική σειρά*

- Προστασία απέναντι σε «ενεργές» επιθέσεις:

  – αλλοίωση των δεδομένων ή των συναλλαγών

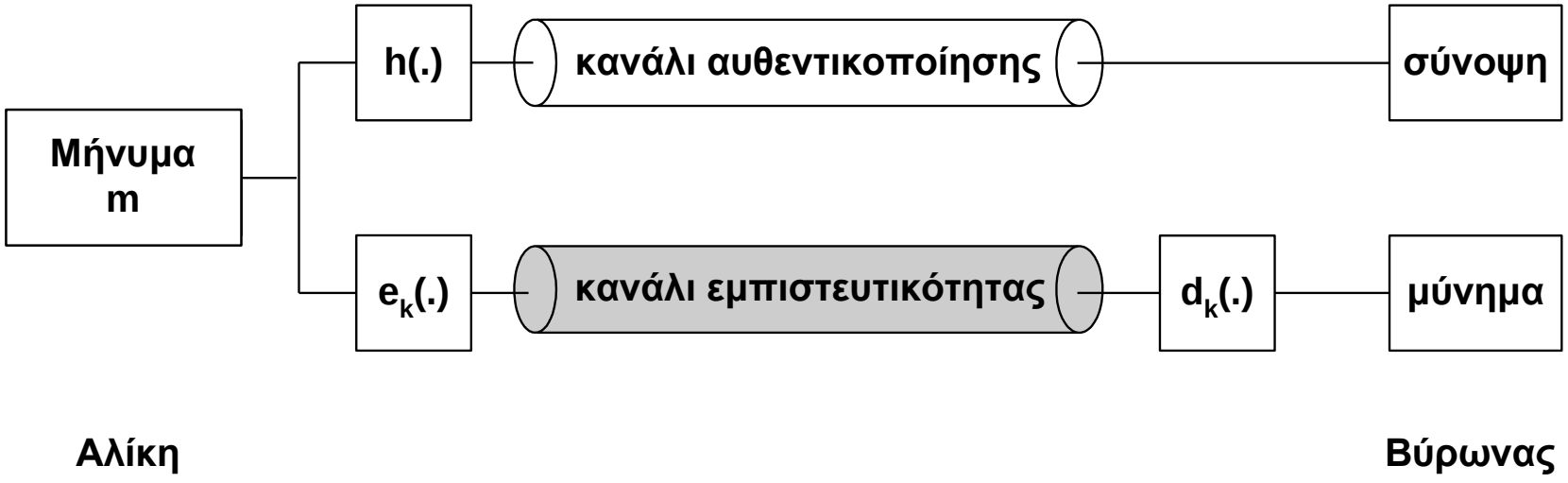
ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗ ΜΗΝΥΜΑΤΟΣ

- ***Κώδικας Αυθεντικοποίησης μηνύματος (Message Authentication Code-MAC)*** είναι μια μονόδρομη hash συνάρτηση με κλειδί η οποία προσφέρει ασθενή αντίσταση σε συγκρούσεις
  - Δοθέντων x, h(x) είναι υπολογιστικά αδύνατο να βρεθεί x' τέτοιο ώστε να ισχύει $h_k(x)=h_k(x')$.



ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗ και ΕΜΠΙΣΤΕΥΤΙΚΟΤΗΤΑ

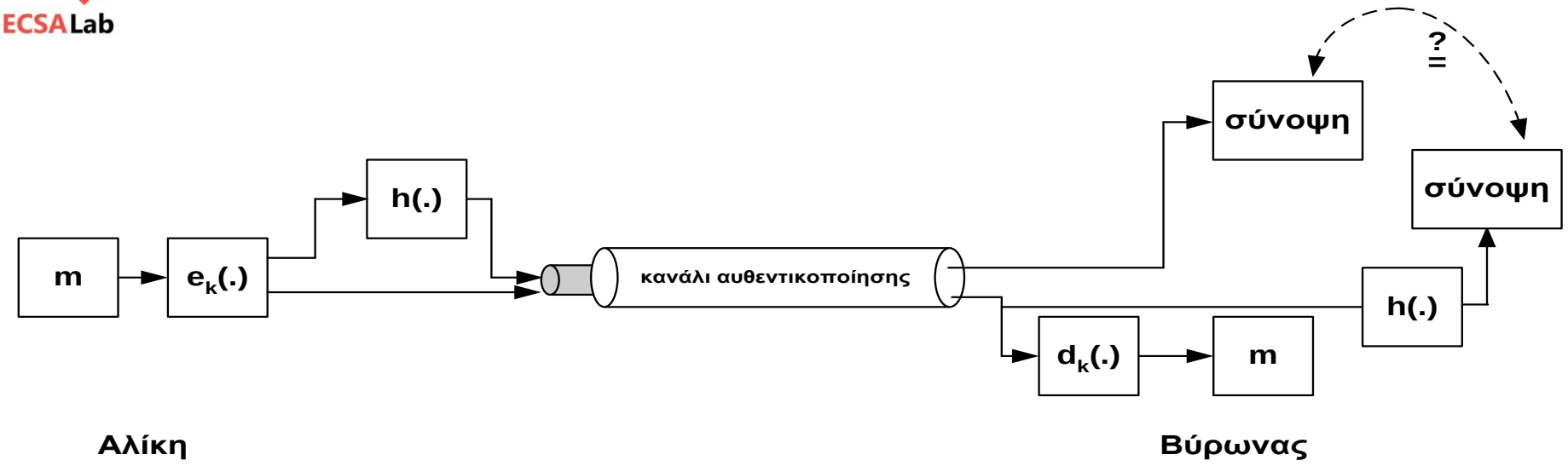
- Για να πετύχουμε αυθεντικοποίηση και εμπιστευτικότητα χρειαζόμαστε δύο συστατικά: Μια μονόδρομη hash που δημιουργεί ένα κανάλι αυθεντικοποίησης και έναν αλγόριθμο κρυπτογραφίας που δημιουργεί ένα κανάλι εμπιστευτικότητας



ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗ ΑΠΛΟΥ ΚΕΙΜΕΝΟΥ

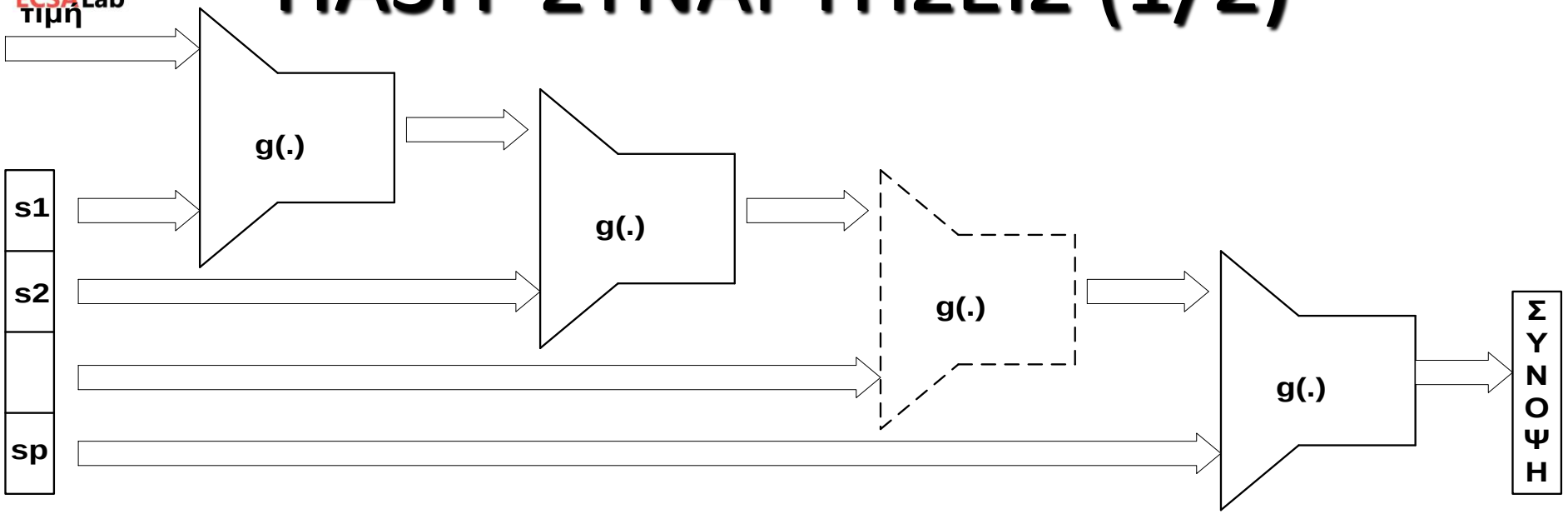
- Η αυθεντικοποίηση πραγματοποιείται στο απλό κείμενο
- Η σύνοψη και το απλό κείμενο κρυπτογραφούνται
- Το κανάλι εμπιστευτικότητας περικλείει το κανάλι αυθεντικοποίησης
  - Πρέπει να αποκρυπτογραφηθούν το μήνυμα και η σύνοψη
  - Ακολουθεί ο έλεγχος αυθεντικοποίησης

ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗ ΚΡΥΠΤΟΚΕΙΜΕΝΟΥ



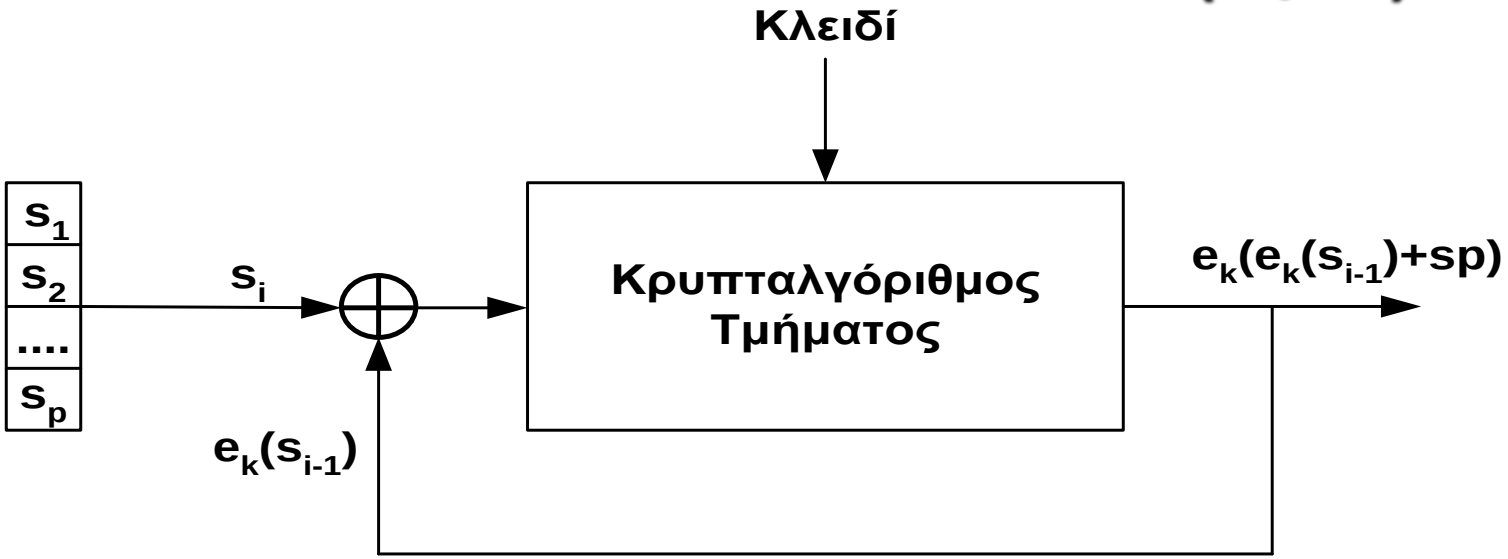
- Η διάταξη αυτή παράγει την σύνοψη του κρυπτοκειμένου
- Η εμπιστευτικότητα προσφέρεται μόνο στο μήνυμα
- Το κανάλι εμπιστευτικότητας βρίσκεται μέσα στο κανάλι αυθεντικοποίησης
  - Ο έλεγχος αυθεντικοποίησης πραγματοποιείται στο λαμβανόμενο κρυπτοκείμενο
  - Σε περίπτωση επιτυχίας ακολουθεί αποκρυπτογράφηση

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΜΟΝΟΔΡΟΜΕΣ HASH ΣΥΝΑΡΤΗΣΕΙΣ (1/2)



- Βασίζονται στην επαναληπτική εφαρμογή μιας συνάρτησης *g*
- *g*: Συνάρτηση συμπίεσης
- Αν η συνάρτηση g είναι ανθεκτική σε συγκρούσεις τότε η επαναληπτική μονόδρομη συνάρτηση θα είναι επίσης ανθεκτική σε συγκρούσεις

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΜΟΝΟΔΡΟΜΕΣ HASH ΣΥΝΑΡΤΗΣΕΙΣ (2/2)



- Χρησιμοποιείται αλγόριθμος τμήματος με είσοδο και έξοδο n-bit
- Το ενδιάμεσο αποτέλεσμα προστίθεται στο επόμενο τμήμα του μηνύματος

ΜΕΓΕΘΟΣ ΣΥΝΟΨΗΣ

- Η επιλογή ενός επαρκούς μεγέθους της σύνοψης για να αποτρέψει τις συγκρούσεις είναι ένα σημαντικό βήμα στον καθορισμό μιας συνάρτησης hash
- 256-bit είναι η αποδεκτή τιμή για την αποφυγή συγκρούσεων

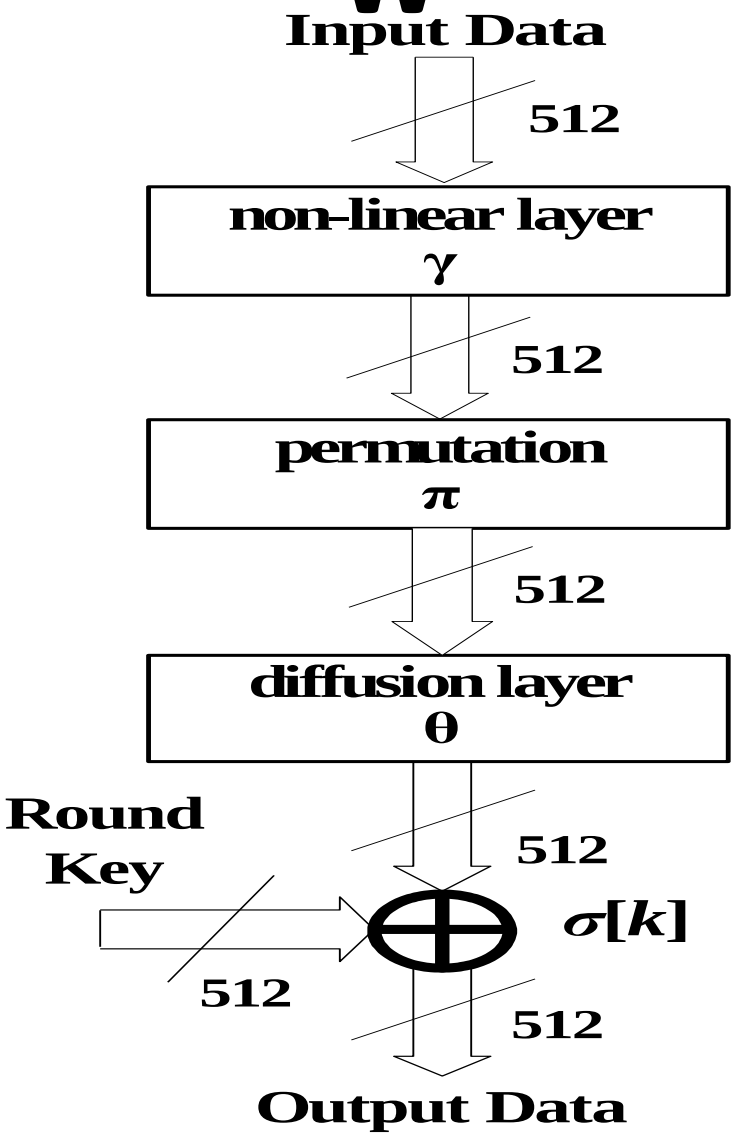
ΠΑΡΑΜΕΤΡΟΙ HASH FUNCTIONS: SHA-1, MD5, WHIRLPOOL

	SHA-1	MD5	Whirlpool
Digest Length	160-bit	128-bit	512-bit
Basic Unit of Processing	512-bit	512-bit	512-bit
Number of Steps	80 (4 x 20)	64 (4 x 16)	10
Maximum Message Size	$2^{64}-1$ bits	+oo	+oo

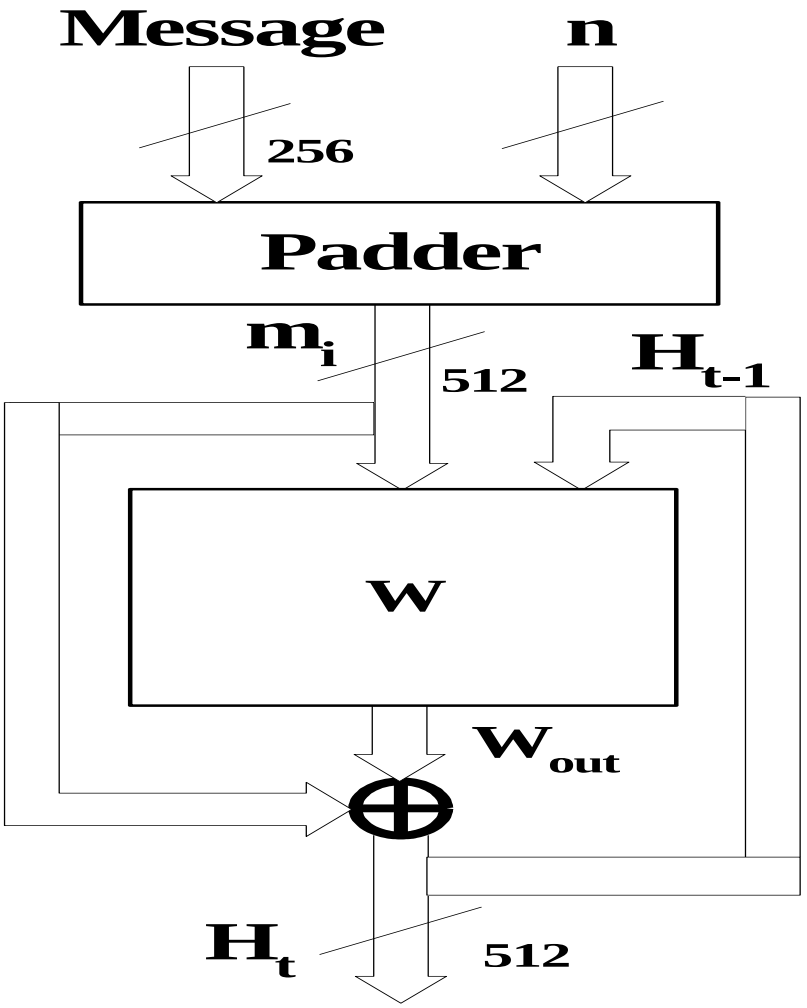
ΣΥΝΑΡΤΗΣΗ ΚΑΤΑΚΕΡΜΑΤΙΣΜΟΥ WHIRLPOOL

- Είναι μέρος του προτύπου ISO/IEC 10118-3 των συναρτήσεων κατακερματισμού
- Είναι μονόδρομη με μέγεθος επεξεργασίας δεδομένων 512-bit
- Αποτελείται από τον αλγόριθμο τμήματος W των 512-bit, με κλειδί 512-bit
- Ο W εκτελείται για 10 γύρους

ΓΥΡΟΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ ΤΜΗΜΑΤΟΣ W

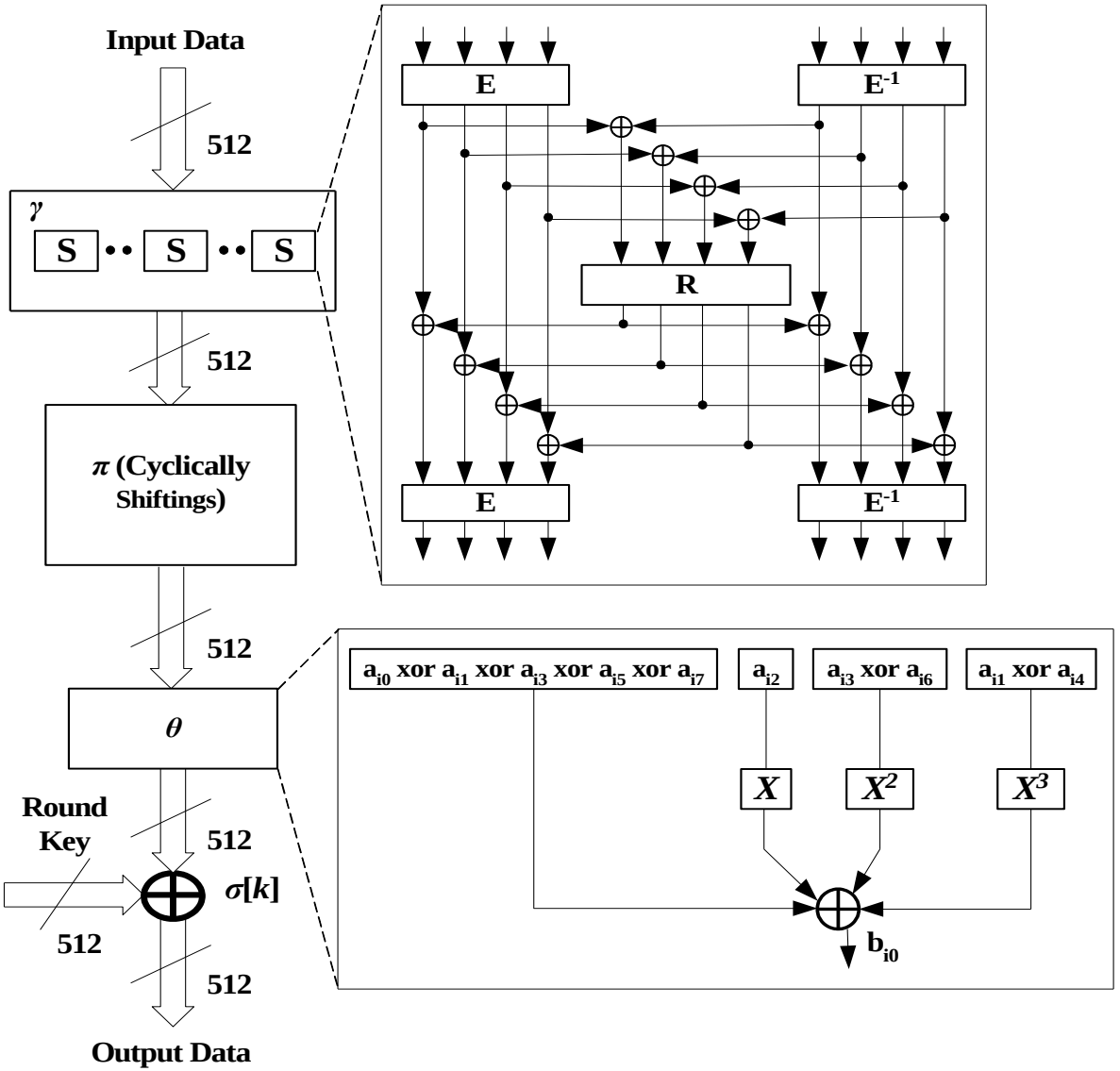


ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΗΣ WHIRLPOOL

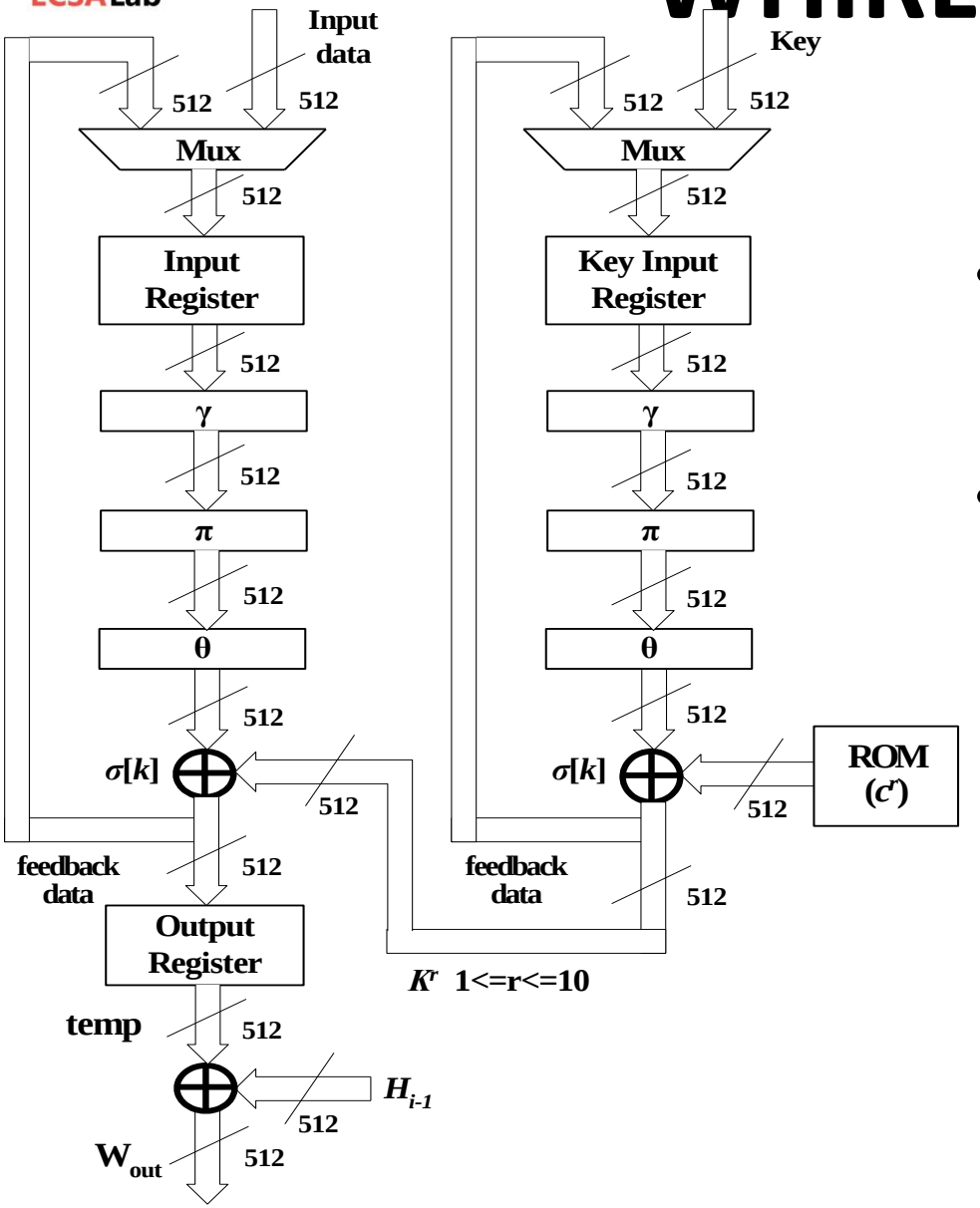


- Ο Padder δημιουργεί ένα μήνυμα μήκους πολλαπλάσιου του 512-bit
- Η είσοδος n, ορίζει το μήκους του μηνύματός

Ο ΓΥΡΟΣ ΤΟΥ W


Input Data
512
γ
S
S
S
512
π (Cyclically Shiftings)
512
θ
Round Key
512
σ[k]
512
512
Output Data
E
E-1
R
E
E-1
ai0 xor ai1 xor ai3 xor ai5 xor ai7
ai2
ai3 xor ai6
ai1 xor ai4
X
X2
X3
bi0

ΣΥΝΟΛΙΚΗ ΥΛΟΠΟΙΗΣΗ ΤΗΣ WHIRLPOOL



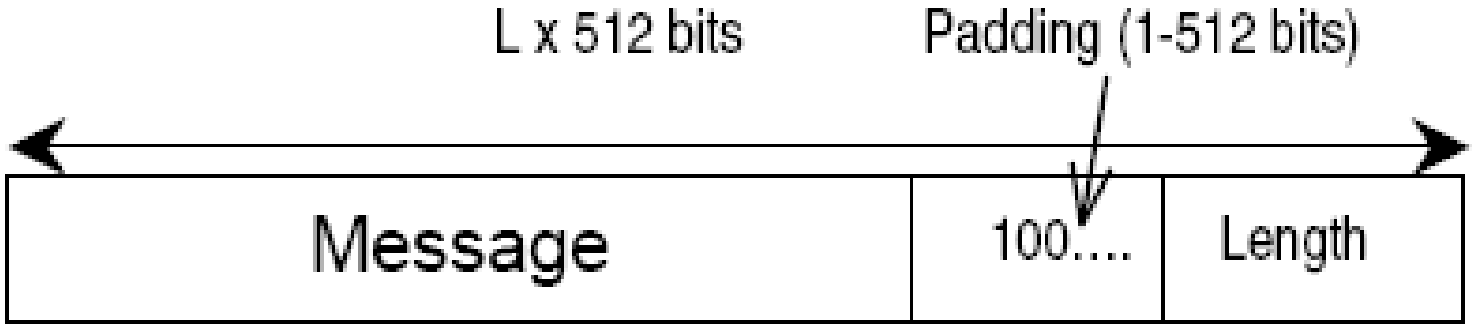
- Αποτελείται από δύο παράλληλους κλάδους επεξεργασίας δεδομένων
- Η είσοδος δεδομένων, mi, εφαρμόζεται ταυτόχρονα με το IV στην είσοδο κλειδιού

MD5 ΣΥΝΑΡΤΗΣΗ (1/6)

- Είσοδος: ένα μήνυμα αυθαίρετου μήκους
- Έξοδος: σύνοψη μήκους 128-bit
- Η επεξεργασία γίνεται σε τμήματα των 512-bit
- Κάθε τμήμα συμμετέχει σε τέσσερις γύρους της συνάρτησης συμπίεσης

MD5 ΣΥΝΑΡΤΗΣΗ (2/6)

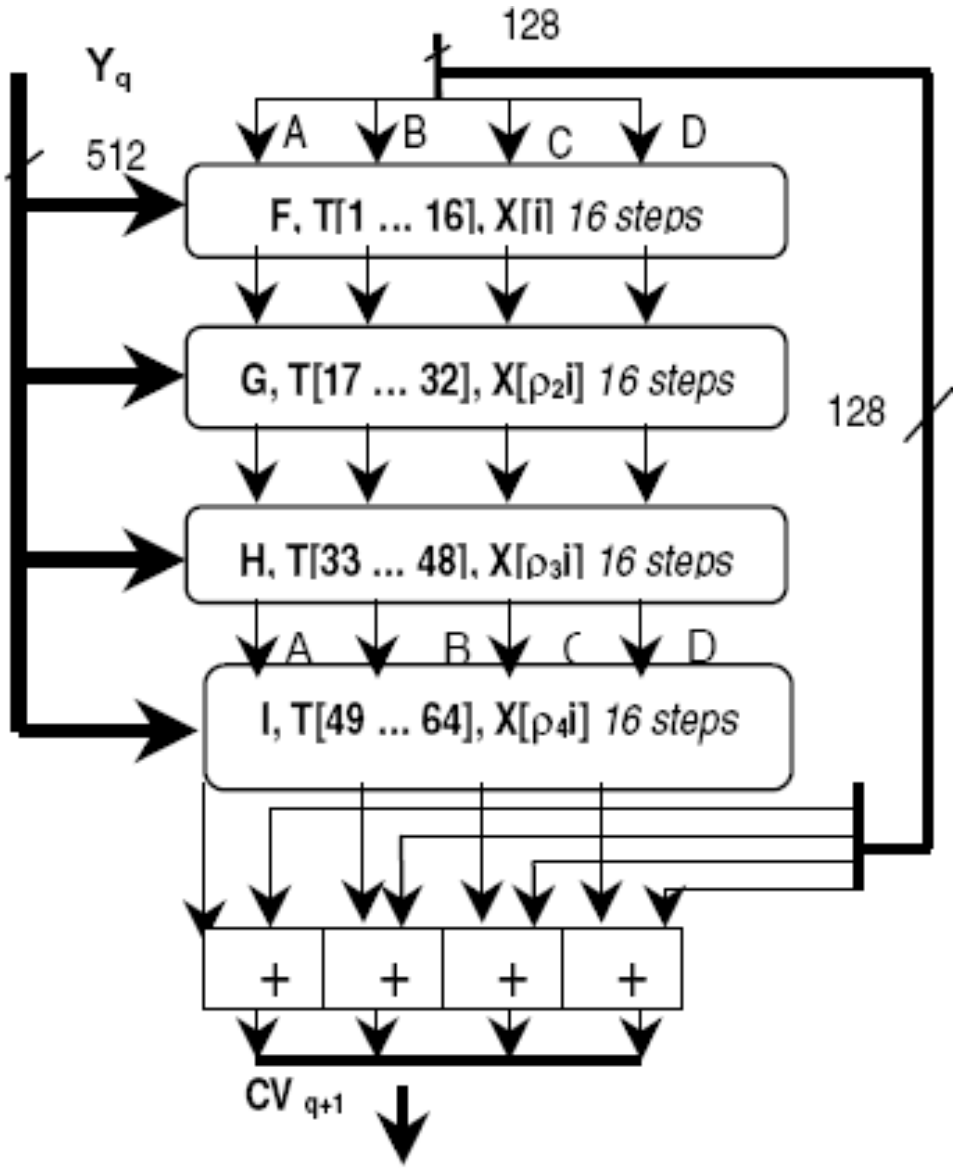
- Στο μήνυμα προστίθενται τόσα bits ώστε το μέγεθός του να είναι ίσο με 448 mod 512
  - Δηλαδή να είναι πολλαπλάσιο του 512
  - Το προστιθέμενο κομμάτι αποτελείται από μια μονάδα και τον απαιτούμενο αριθμό μηδενικών
- Τα τελευταία 64-bit χρησιμοποιούνται για να αποθηκευτεί ο αριθμός που εκφράζει το συνολικό μήκος του μηνύματος



MD5 ΣΥΝΑΡΤΗΣΗ (3/6)

- Οι ενδιάμεσες τιμές καθώς και το αποτέλεσμα της σύνοψης αποθηκεύονται σε 4 καταχωρητές, που αρχικά έχουν τις παρακάτω τιμές:
- $A=(67452301)_{16}$
- $B=(EFCDAB89)_{16}$
- $C=(98BADCFE)_{16}$
- $D=(19325476)_{16}$
- Η συνάρτηση συμπίεσης αποτελείται από 4 γύρους και κάθε γύρος εκτελεί πράξεις για 16 φορές
- Κάθε πράξη εκτελεί μία μη γραμμική συνάρτηση μεταξύ των τριών από τα A, B, C, D και προσθέτει το αποτέλεσμα στην τέταρτη μεταβλητή

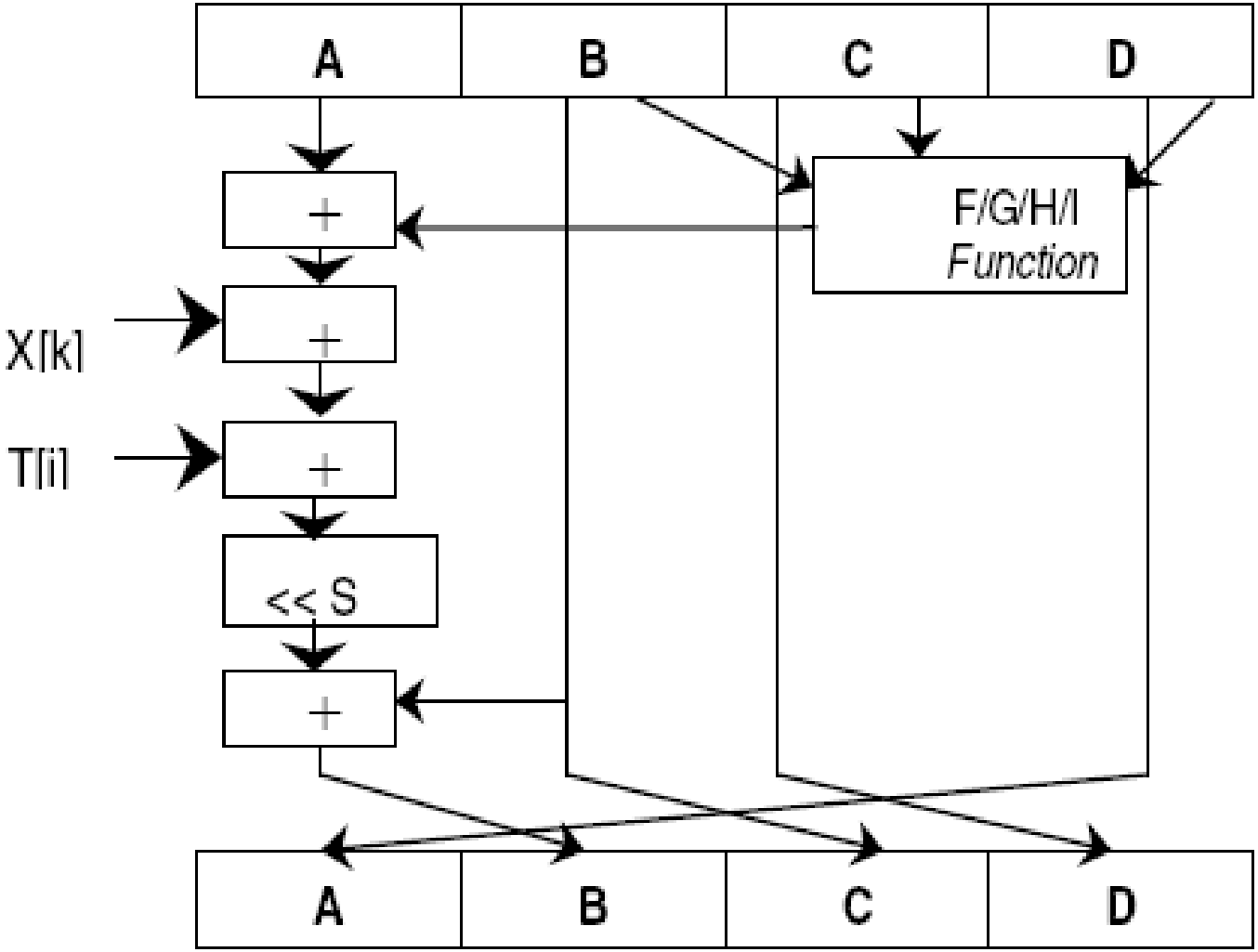
MD5 ΣΥΝΑΡΤΗΣΗ (4/6)



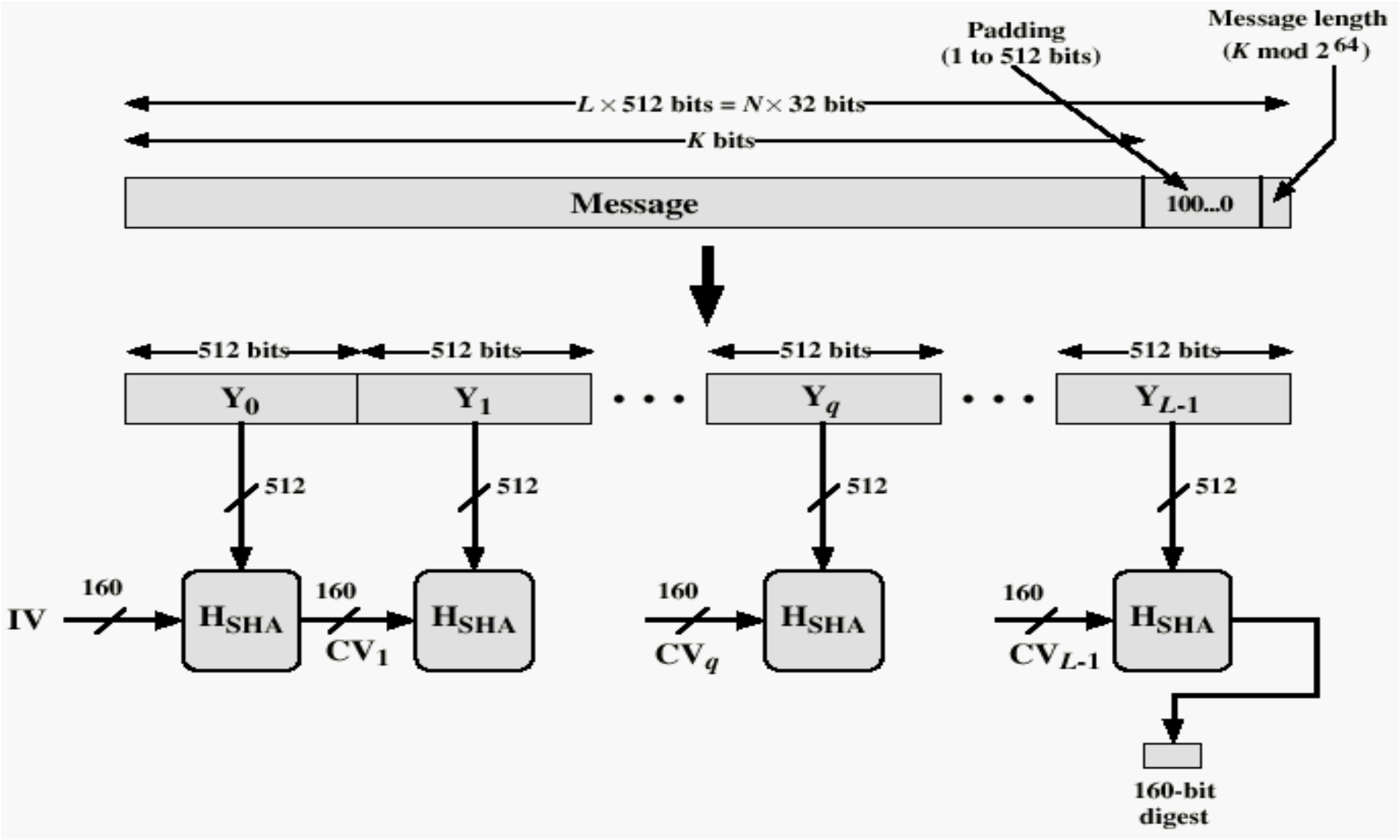
MD5 ΣΥΝΑΡΤΗΣΗ (5/6)

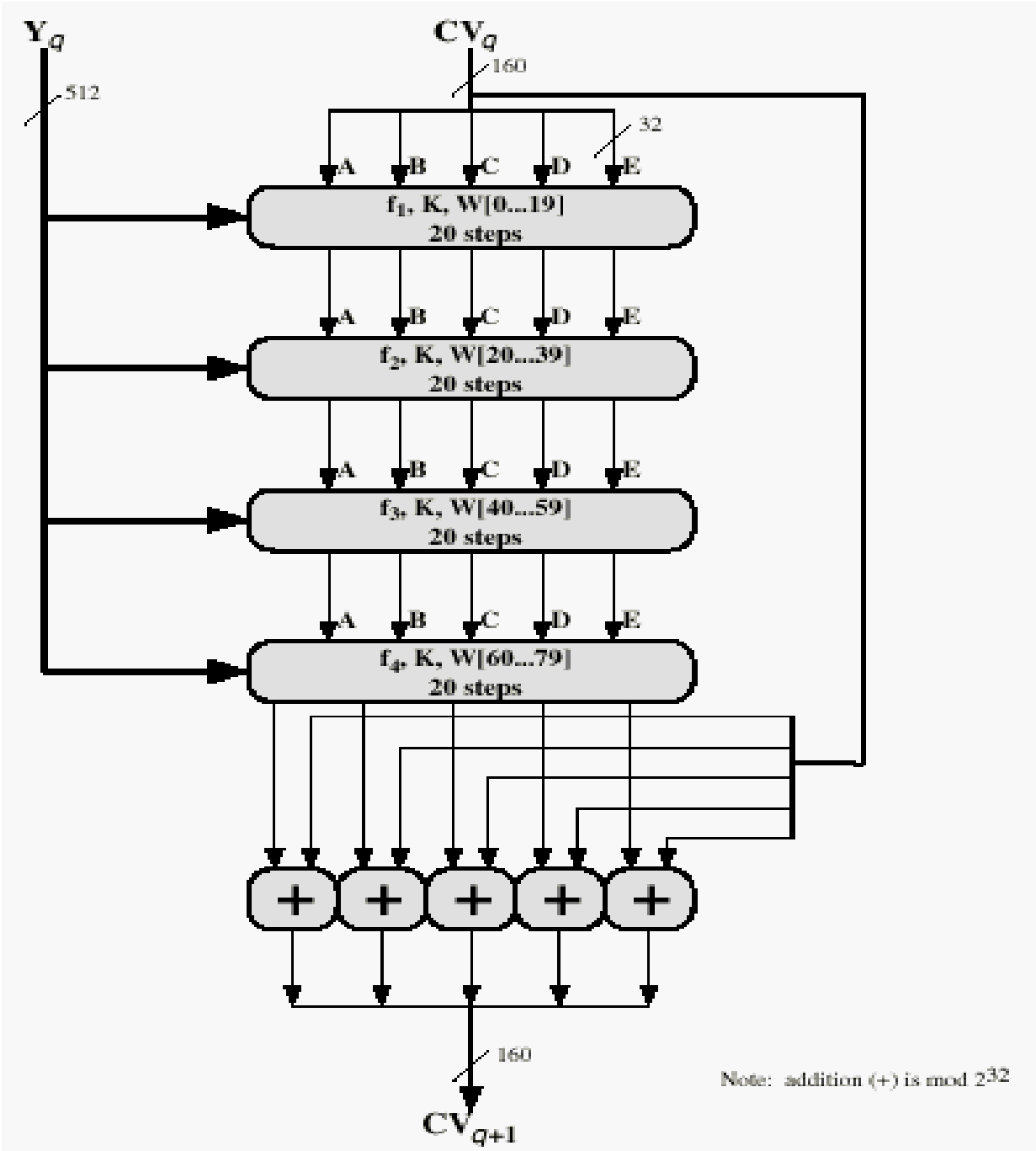
- Στο τέλος των τεσσάρων γύρων στις τελικές τιμές προστίθενται οι αρχικές τιμές των μεταβλητών A, B, C και D
- Χρησιμοποιεί 4 βοηθητικές συναρτήσεις όπου κάθε μια παίρνει σαν είσοδο τρεις λέξεις των 32-bit και επιστρέφει μία λέξη των 32-bit
- F(X,Y,Z)=XY vnot(X)Z
- G(X,Y,Z)=XZ vY not(Z)
- H(X,Y,Z)=X ⊕Y ⊕Z
- I(X,Y,Z)=Y ⊕(X vnot(Z))
- Επίσης χρησιμοποιεί ένα πίνακα 64άρων στοιχείων T

MD5 ΣΥΝΑΡΤΗΣΗ (6/6)



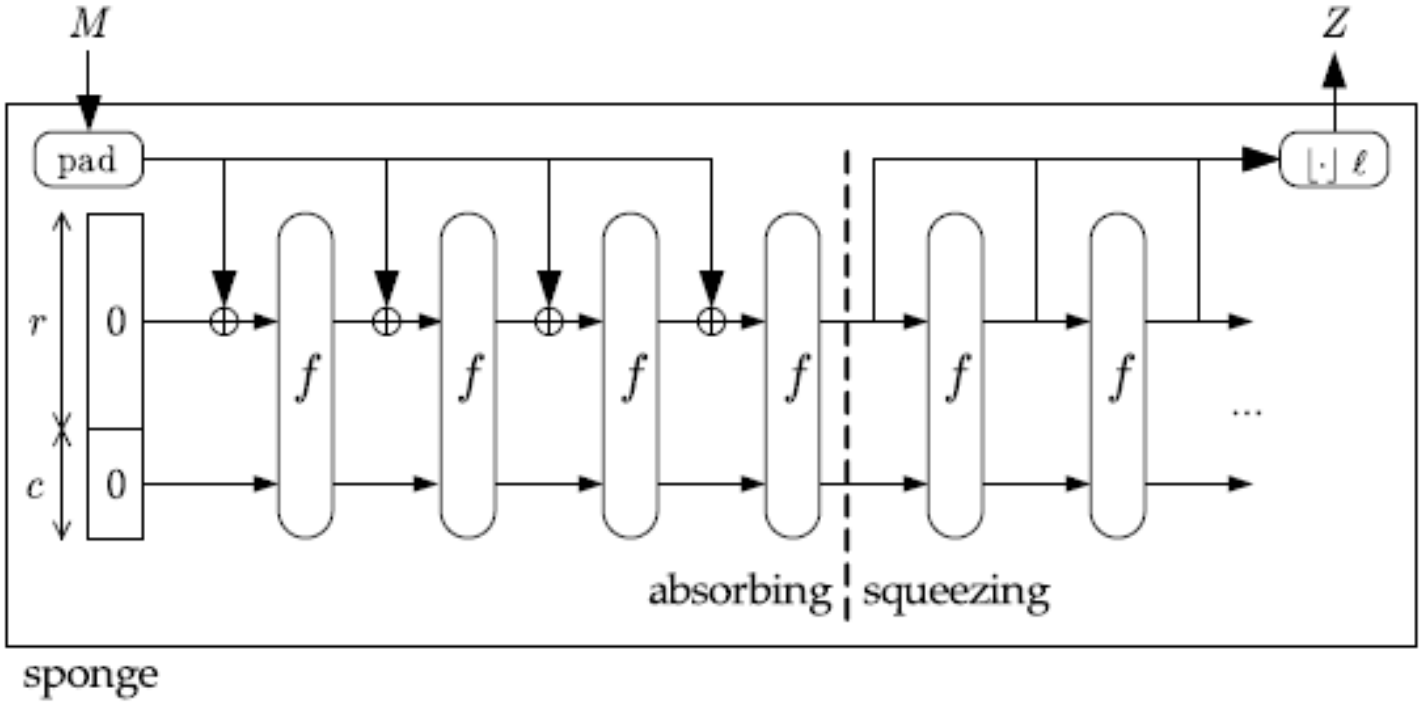
SHA-1: ΔΗΜΙΟΥΡΓΙΑ MESSAGE DIGEST




Y_q
512
CV_q
160
32
A
B
C
D
E
f_1, K, W[0...19]
20 steps
f_2, K, W[20...39]
20 steps
f_3, K, W[40...59]
20 steps
f_4, K, W[60...79]
20 steps
+
160
CV_{q+1}
Note: addition (+) is mod 2^{32}

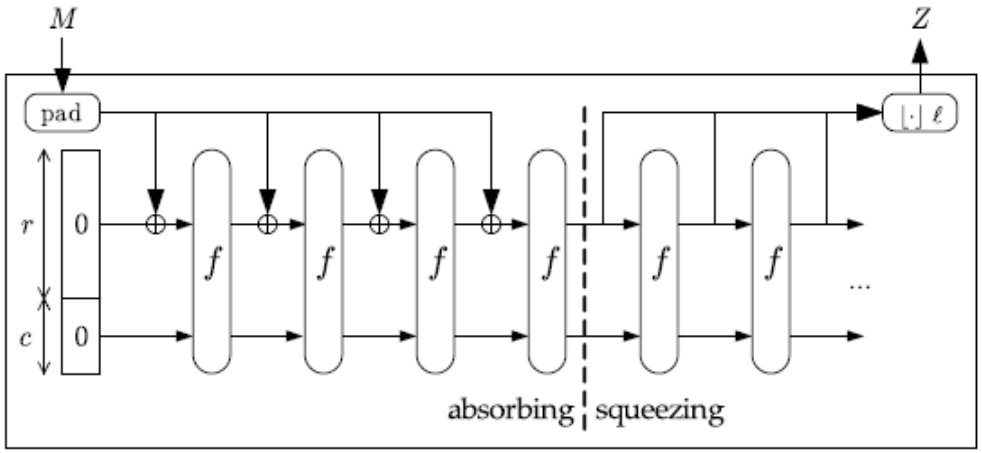
KECCAK HASH FUNCTION (1/8)

- Η Keccak χρησιμοποιεί ένα οικοδόμημα «σπόγγου – sponge» κατά το οποίο τα δεδομένα εισόδου «απορροφώνται – absorbed» από τον σπόγγο
- Και το δεδομένα εξόδου συμπιέζονται «στύβονται – squeezed» από τον σπόγγο

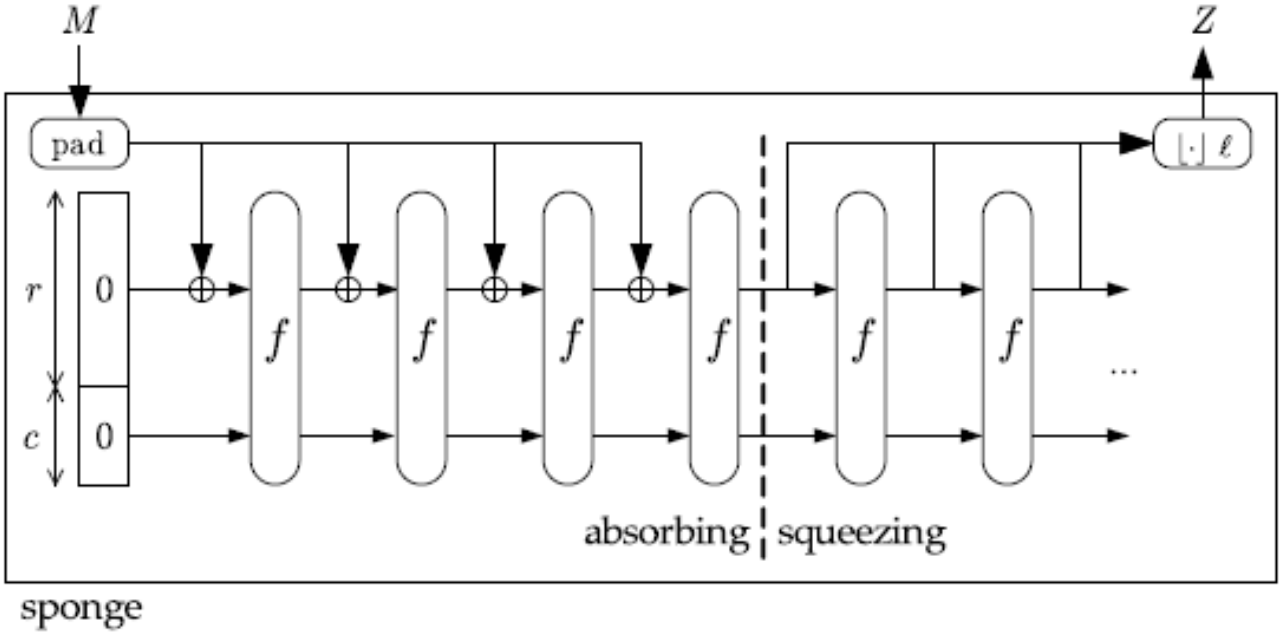


KECCAK HASH FUNCTION (2/8)

- Κατά τη φάση «απορρόφησης», μέρος της εισόδου (state) των δεδομένων εκτελεί XOR με ένα διάνυσμα αρχικοποίησης, το οποίο μετά μετασχηματίζεται στο σύνολό της χρησιμοποιώντας μια συνάρτηση f
- Κατά τη φάση «συμπίεσης», τα μπλοκ εξόδου διαβάζονται από διαδοχικές εξόδους της συνάρτησης f του ίδιου υποσύνολο της state



KECCAK HASH FUNCTION (3/8)

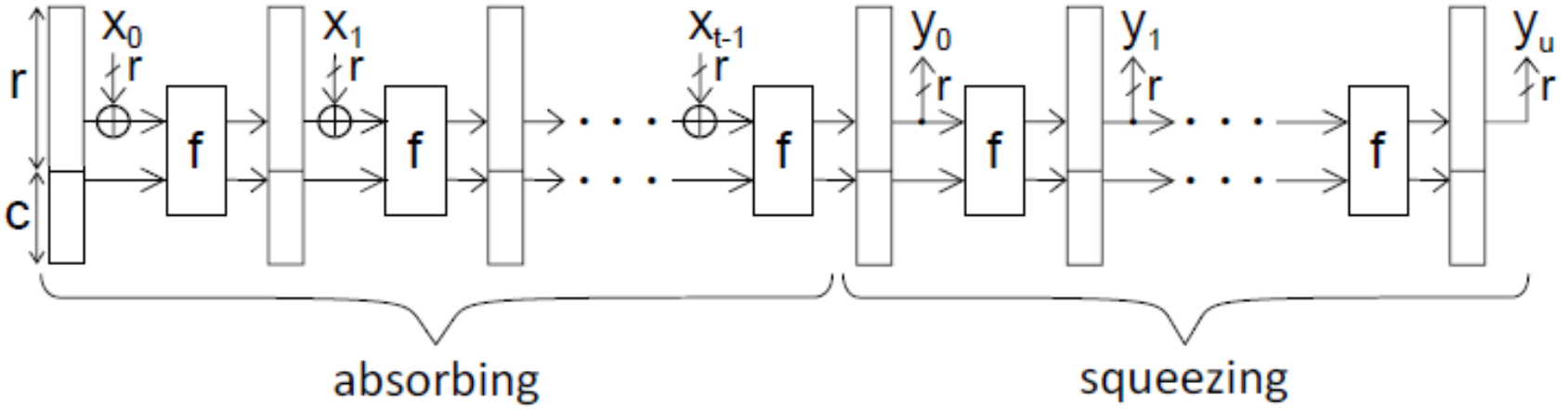


- M: message (input)
- c: capacity
- r : rate
- Z: digest (output)
- l is the output length
- f: permutation of b = r + c bits

KECCAK HASH FUNCTION (4/8)

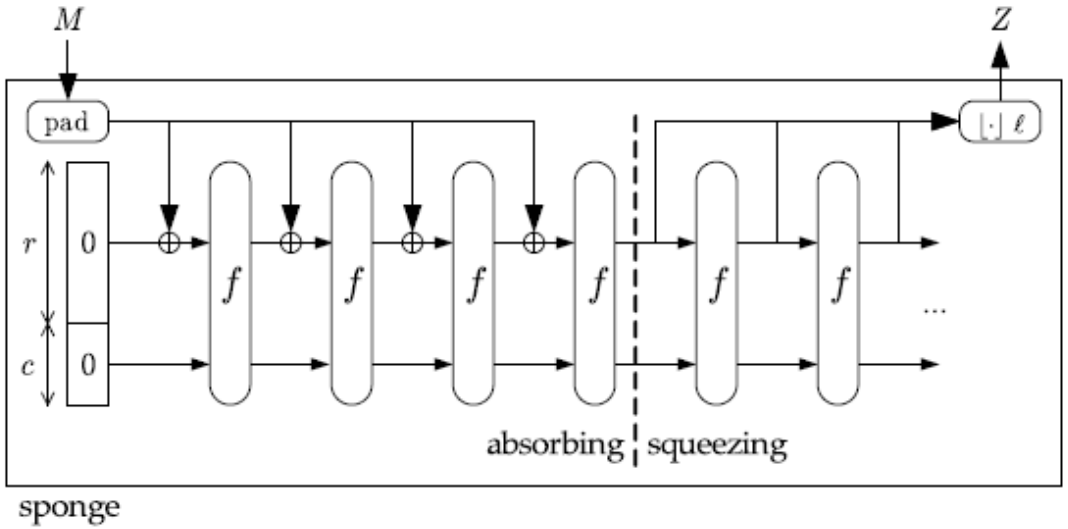
- Το μήκος του μέρους του state το οποίο χρησιμοποιείται για την ανάγνωση και τη καταγραφή καλείται «rate» και δηλώνεται ως r,
- Το μήκος του μέρους του state το οποίο αποκολλάται στην είσοδο και την έξοδο καλείται «capacity» και δηλώνεται ως c
  - Το μήκος του capacity καθορίζει το επίπεδο ασφάλειας της συνάρτησης

KECCAK HASH FUNCTION (5/8)



- Στην είσοδο γίνεται padding έτσι ώστε να γίνει (η είσοδος) πολλαπλάσιο του r-bit (pad10*1(*r, M)*)
- Φάση απορρόφησης: Σε κάθε γύρο r-bit του μηνύματος υφίσταται XOR με r-bit της κατάστασης
- Όταν ολοκληρωθεί η επεξεργασία όλης της εισόδου περνάμε στη συμπίεση
- Στη φάση συμπίεσης, σε κάθε γύρο r-bit της κατάστασης εξέρχονται ως έξοδο ώστε να δημιουργηθεί ένα μήνυμα μήκους l

KECCAK HASH FUNCTION (6/8)



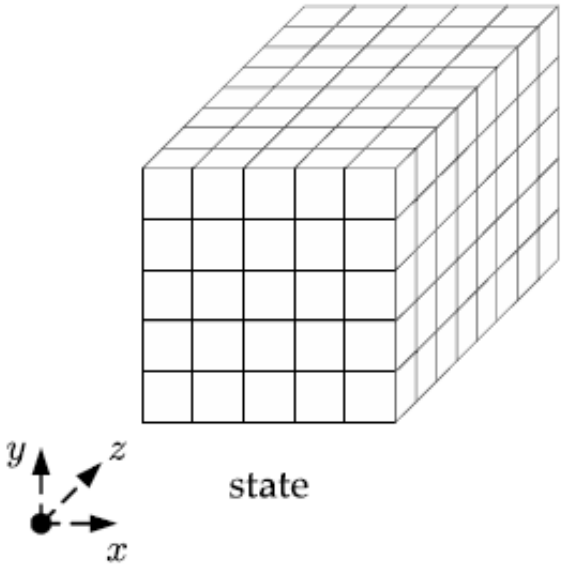
Name	ℓ	c	r
SHA3-224	224	448	1152
SHA3-256	256	512	1088
SHA3-384	384	768	832
SHA3-512	512	1024	576

- Ισχύει ότι c + r = 1600

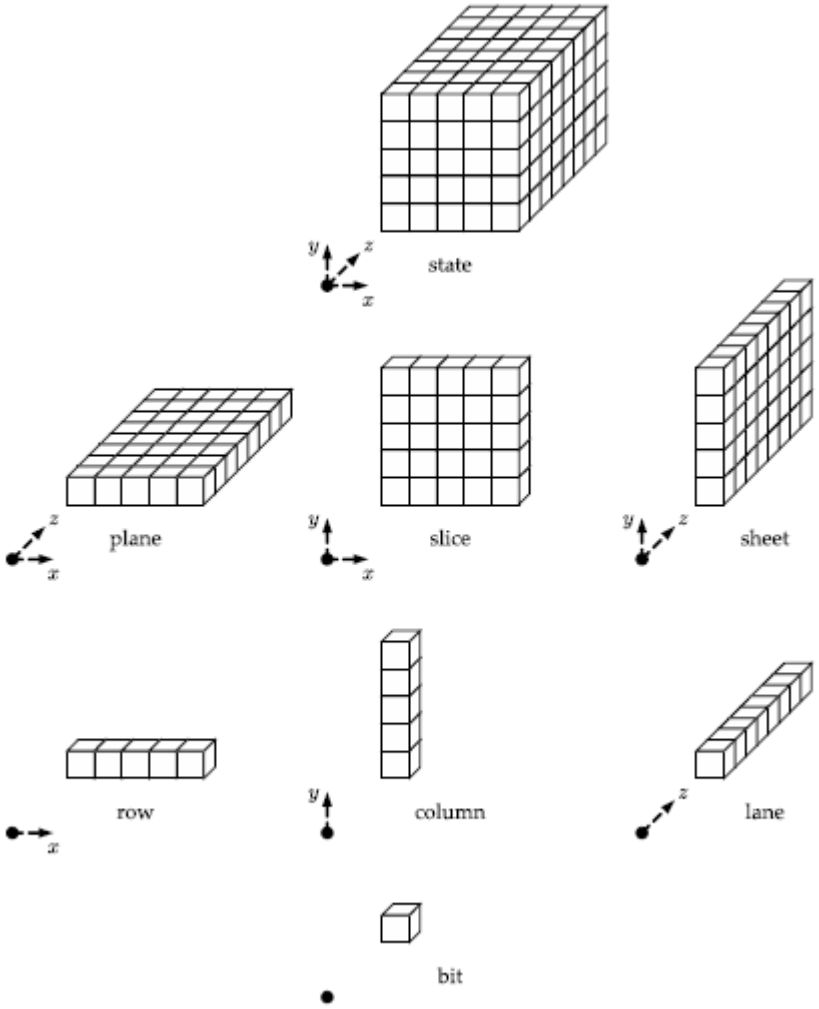
KECCAK HASH FUNCTION (7/8)

Περιγραφή **state**

- Περιγράφει τα 1600-bit εισόδου στην συνάρτηση f
- Είναι ένας κύβος 5 x 5 x 64-bit
- Κάθε θέση (τετράγωνο) είναι 1-bit
- $0 \leq x, y \leq 4, 0 \leq z \leq 63$
- Μπορεί να αναπαρασταθεί ως μια δομή από 25 ακεραίους

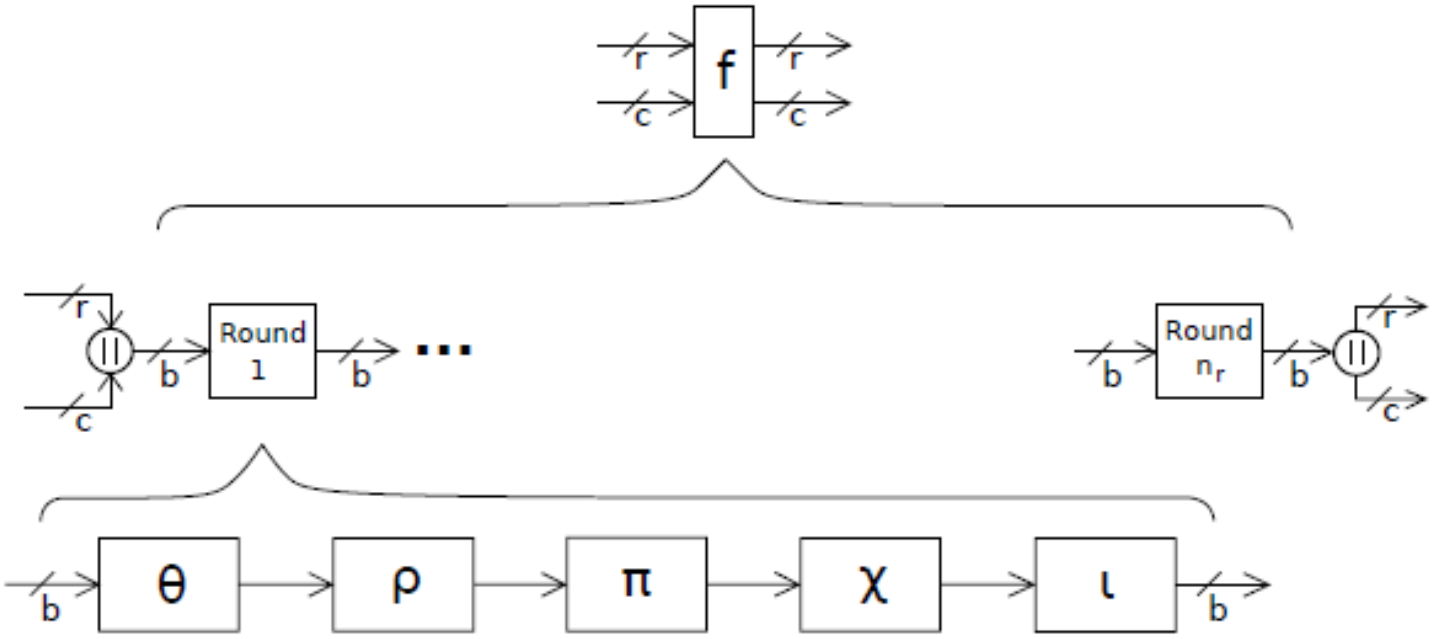


KECCAK HASH FUNCTION (8/8)



ΣΥΝΑΡΤΗΣΗ ΓΥΡΟΥ (f)

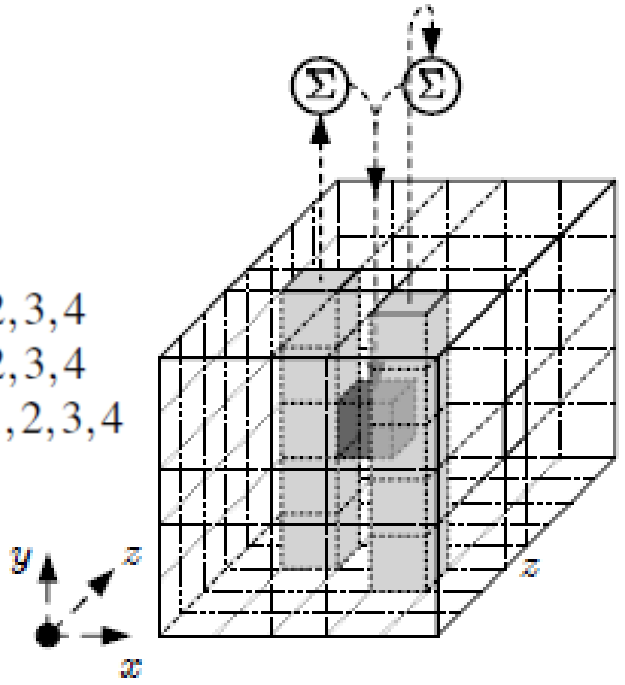
- Η συνάρτηση f αποτελείται από 24 rounds (γύρους)
- 1 γύρος: Δεδομένου ενός state A και τον δείκτη του round ir,
  $Rnd(A,ir) = \iota_{ir} \circ \chi \circ \pi \circ \rho \circ \Theta(A)$



ΣΥΝΑΡΤΗΣΗ ΤΗΙΤΑ (Θ)

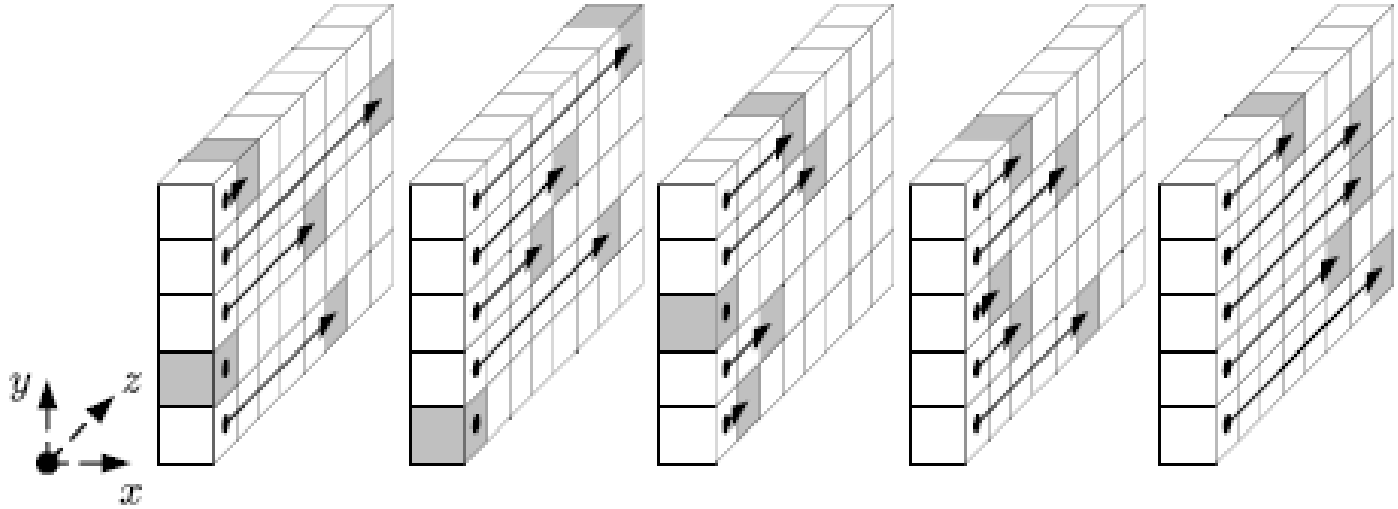
- Άθροισμα όλων των στηλών
- Αθροίζει το άθροισμα δύο στηλών
  (x - 1; z) (x + 1; z - 1) σε
  τιμή ενός bit (x; y; z)

$$
\begin{aligned}
C[x] &= A[x,0] \oplus A[x,1] \oplus A[x,2] \oplus A[x,3] \oplus A[x,4] && , x = 0,1,2,3,4 \\
D[x] &= C[x-1] \oplus \mathrm{rot}(C[x+1],1) && , x = 0,1,2,3,4 \\
A[x,y] &= A[x,y] \oplus D[x] && , x,y = 0,1,2,3,4
\end{aligned}
$$



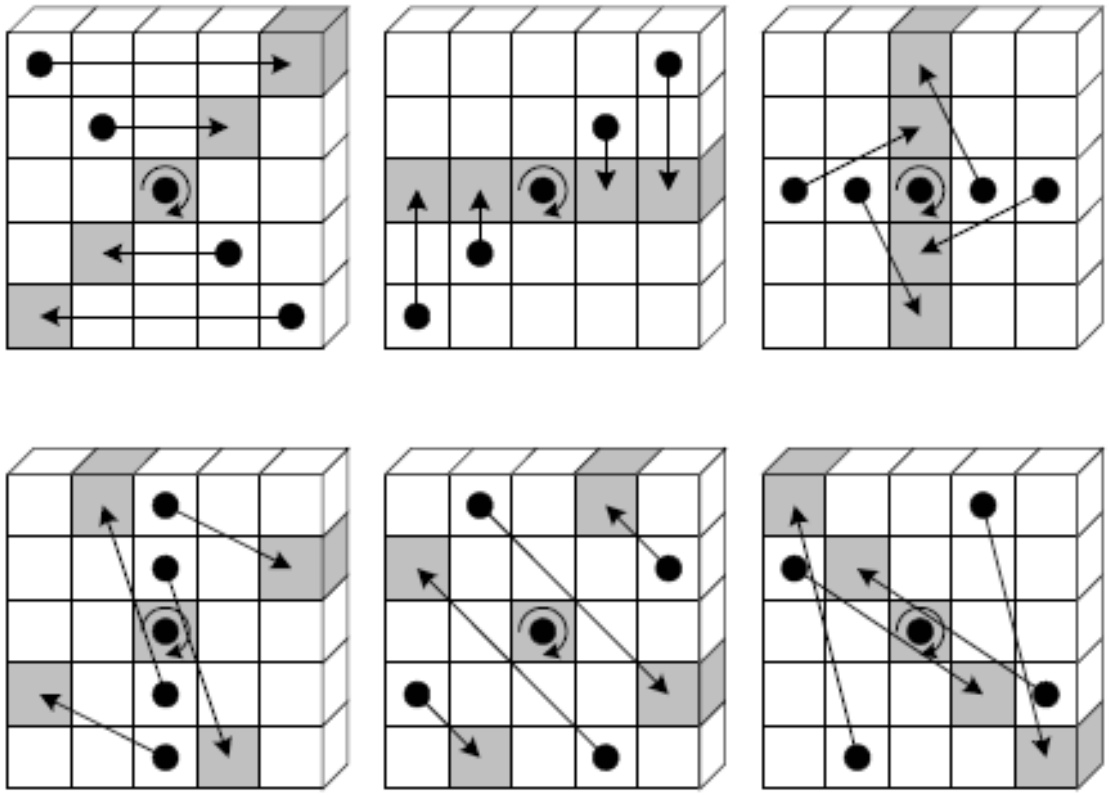
ΣΥΝΑΡΤΗΣΗ RHO (ρ)

- Ολισθαίνει κυκλικά κάθε γραμμή κατά κάποιες θέσεις (25 γραμμές)



ΣΥΝΑΡΤΗΣΗ PI (π)

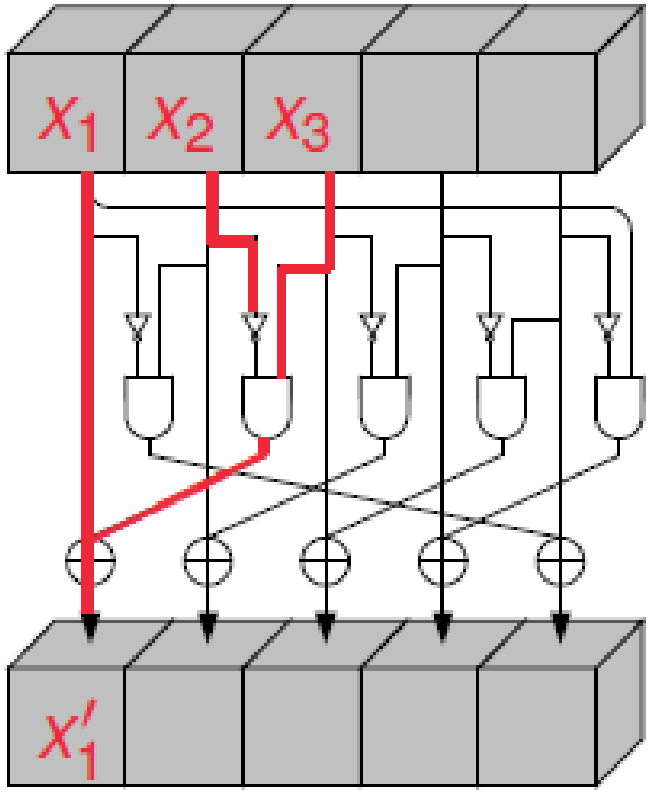
- Αναδιατάσσει τις θέσεις των γραμμών



1. For all triples (x, y, z) such that $0 \le x < 5$, $0 \le y < 5$, and $0 \le z < w$, let
   $\mathbf{A}'[x, y, z] = \mathbf{A}[(x + 3y) \bmod 5, x, z]$.
2. Return $\mathbf{A}'$.

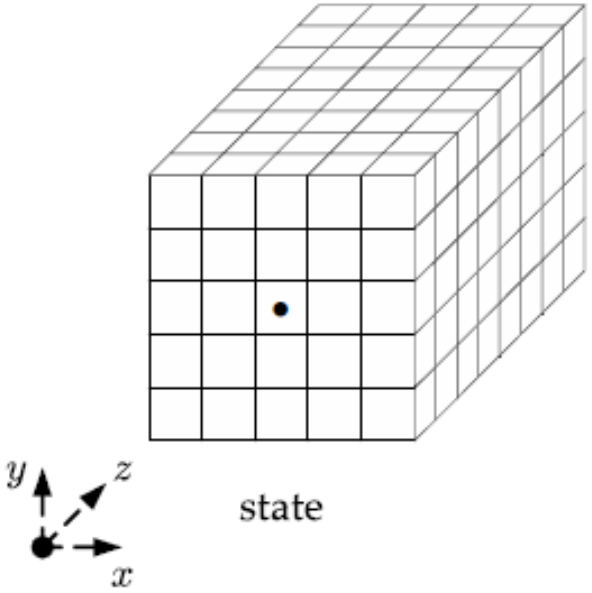
ΣΥΝΑΡΤΗΣΗ CHI (χ)

- Εκτελεί XOR ενός bit με μια μη-γραμμική συνάρτηση μεταξύ δύο άλλων bits στην ίδια γραμμή
- Για παράδειγμα, $x_1' <- x_1 + x_2'x_3$
- Συμπεριφέρεται ως ένα Sbox των 5-bit



ΣΥΝΑΡΤΗΣΗ IOTA (ι)

- Τροποποιεί κάποια από τα bits της γραμμής (0, 0) έτσι ώστε να εξαρτάται από τον δείκτη του γύρου ir
- Εάν οι γραμμές αναπαρίστανται ως ένας ακέραιος των 64-bit, ουσιαστικά είναι η XOR με μια σταθερά γύρου



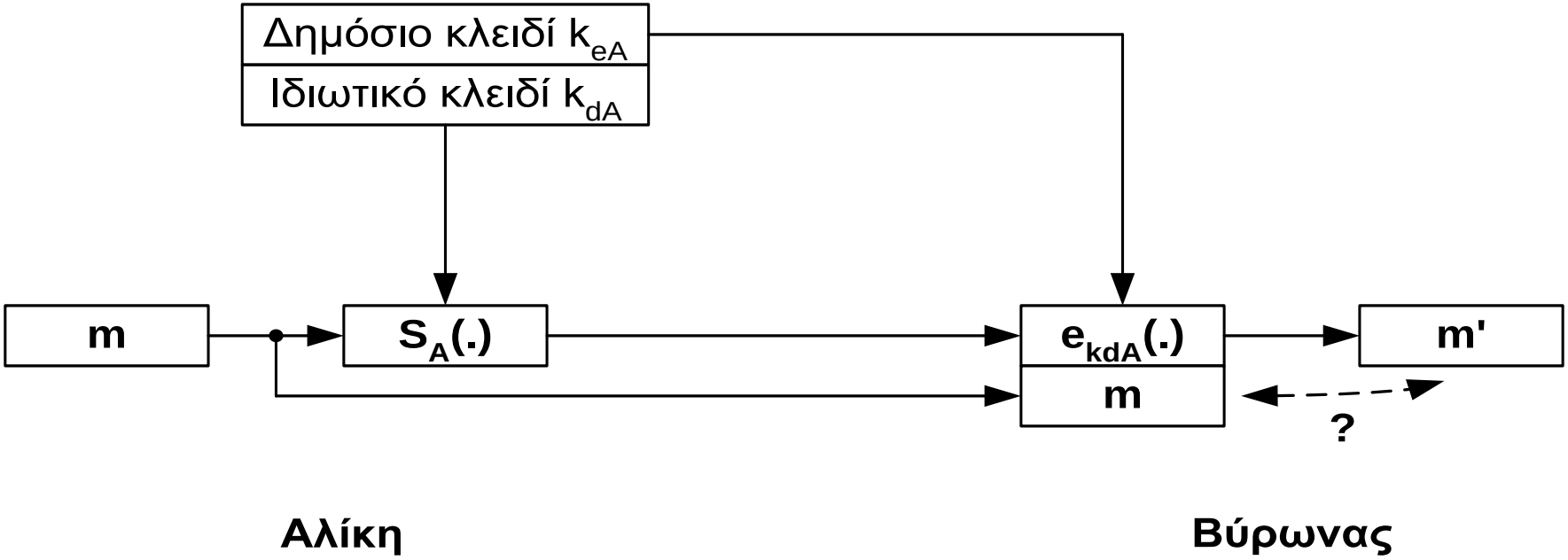
ΣΤΑΘΕΡΕΣ ΓΥΡΟΥ

RC[0] = 0x0000000000000001
RC[1] = 0x0000000000008082
RC[2] = 0x800000000000808A
RC[3] = 0x8000000080008000
RC[4] = 0x000000000000808B
RC[5] = 0x0000000080000001
RC[6] = 0x8000000080008081
RC[7] = 0x8000000000008009
RC[8] = 0x000000000000008A
RC[9] = 0x0000000000000088
RC[10] = 0x0000000080008009
RC[11] = 0x000000008000000A
RC[12] = 0x000000008000808B
RC[13] = 0x800000000000008B
RC[14] = 0x8000000000008089
RC[15] = 0x8000000000008003
RC[16] = 0x8000000000008002
RC[17] = 0x8000000000000080
RC[18] = 0x000000000000800A
RC[19] = 0x800000008000000A
RC[20] = 0x8000000080008081
RC[21] = 0x8000000000008080
RC[22] = 0x0000000080000001
RC[23] = 0x8000000080008008

ΧΡΗΣΕΙΣ ΤΗΣ ΨΗΦΙΑΚΗΣ ΥΠΟΓΡΑΦΗΣ

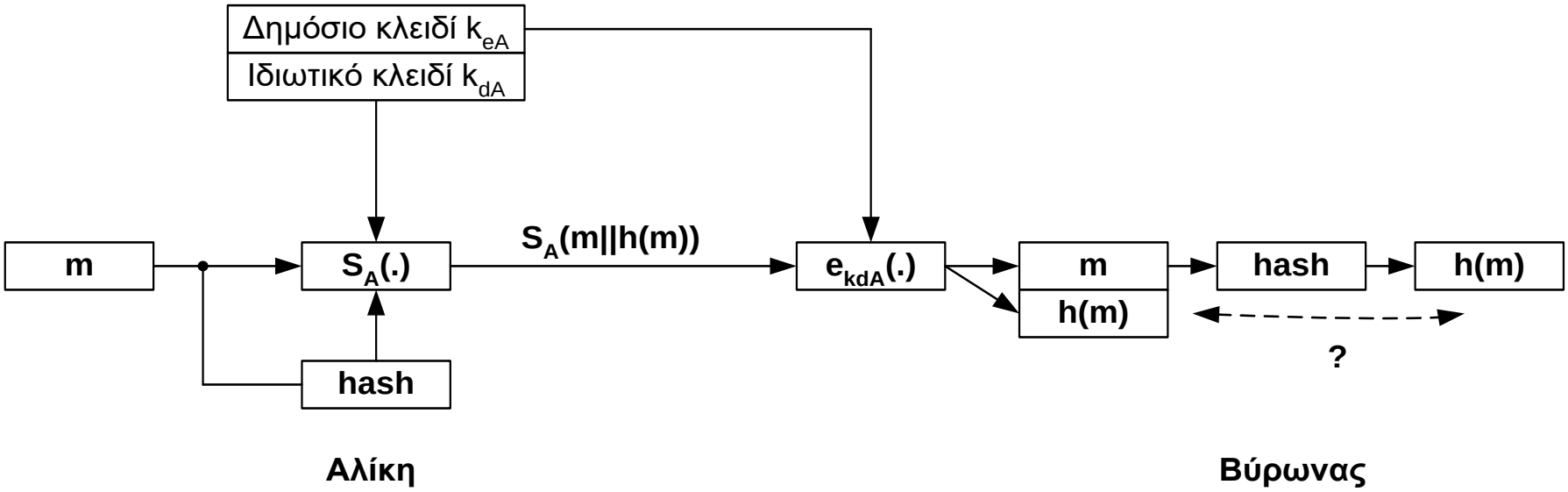
- Αυθεντικοποίηση της πηγής του μηνύματος
  - Προστατεύει τον αποστολέα ακόμα και σε περίπτωση που ο παραλήπτης τροποποιήσει το αρχικό μήνυμα του αποστολέα
- Μη απάρνηση πηγής
  - Σε περίπτωση που ο αποστολέας αρνηθεί ότι έστειλε το μήνυμα, ο παραλήπτης του μηνύματος θα πρέπει να είναι σε θέση να αποδείξει ότι το μήνυμα στάλθηκε από τον αποστολέα
- Μη απάρνηση προορισμού
  - Σε περίπτωση που ο παραλήπτης αρνηθεί ότι παρέλαβε το μήνυμα, θα πρέπει να υπάρχει δυνατότητα απόδειξης ότι το μήνυμα παραλήφθηκε από τον παραλήπτη

ΨΗΦΙΑΚΕΣ ΥΠΟΓΡΑΦΕΣ ΑΣΥΜΜΕΤΡΗΣ ΚΡΥΠΤΟΓΡΑΦΙΑΣ

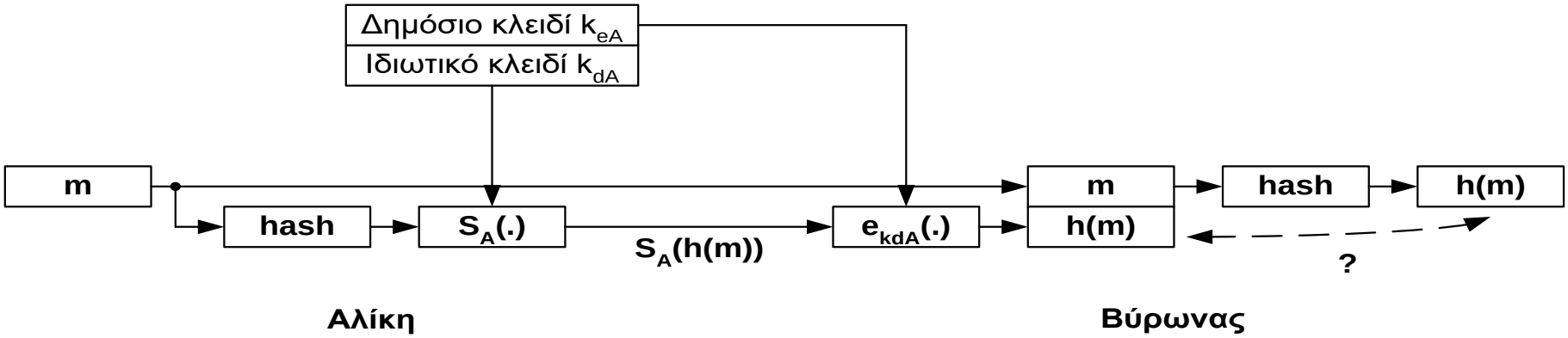


- Ο όγκος των μηνυμάτων που στέλνονται είναι διπλάσιος του μεγέθους του αρχικού μηνύματος

ΣΥΣΤΗΜΑΤΑ ΨΗΦΙΑΚΗΣ ΥΠΟΓΡΑΦΗΣ ΜΕ ΑΥΤΟΑΝΑΚΤΗΣΗ



ΣΥΣΤΗΜΑΤΑ ΨΗΦΙΑΚΗΣ ΥΠΟΓΡΑΦΗΣ ΜΕ ΠΑΡΑΡΤΗΜΑ



ΨΗΦΙΑΚΕΣ ΥΠΟΓΡΑΦΕΣ ΜΕ ΤΟ ΚΡΥΠΤΟΣΥΣΤΗΜΑ RSA

- Το σύστημα αυτό απαιτεί όλες οι οντότητες να έχουν στην κατοχή τους αντίστοιχα ζεύγη δημοσίου και ιδιωτικού κλειδιού.
- Έστω p και q δύο πρώτοι αριθμοί και n=pq. Το σύστημα ψηφιακών υπογραφών RSA ορίζεται με $M=S=Z_n$, πράξη υπογραφής

$$S_A(m) = m^{k_{dA}} \bmod n$$

και πράξη επαλήθευσης

$$V_A(s) = s^{k_{eA}} \bmod n$$

όπου $k_{eA}k_{dA} \equiv 1(\bmod \phi(n))$ με k_{eA} το δημόσιο κλειδί και k_{dA} το ιδιωτικό κλειδί της οντότητας Α.

ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗ ΜΗΝΥΜΑΤΟΣ (MESSAGE AUTHENTICATION)

- Ένα σύστημα θα πρέπει να έχει τη δυνατότητα να πιστοποιεί ότι:

  1. *Ένα μήνυμα έχει αποσταλεί από συγκεκριμένη πηγή ή αποστολέα*
  2. *Τα περιεχόμενα του μηνύματος δεν έχουν διαφοροποιηθεί*
  3. *Το δεδομένα έχουν αποσταλεί σε συγκεκριμένη χρονική περίοδο και με συγκεκριμένη διαδοχική σειρά*

- Προστασία απέναντι σε «ενεργές» επιθέσεις:

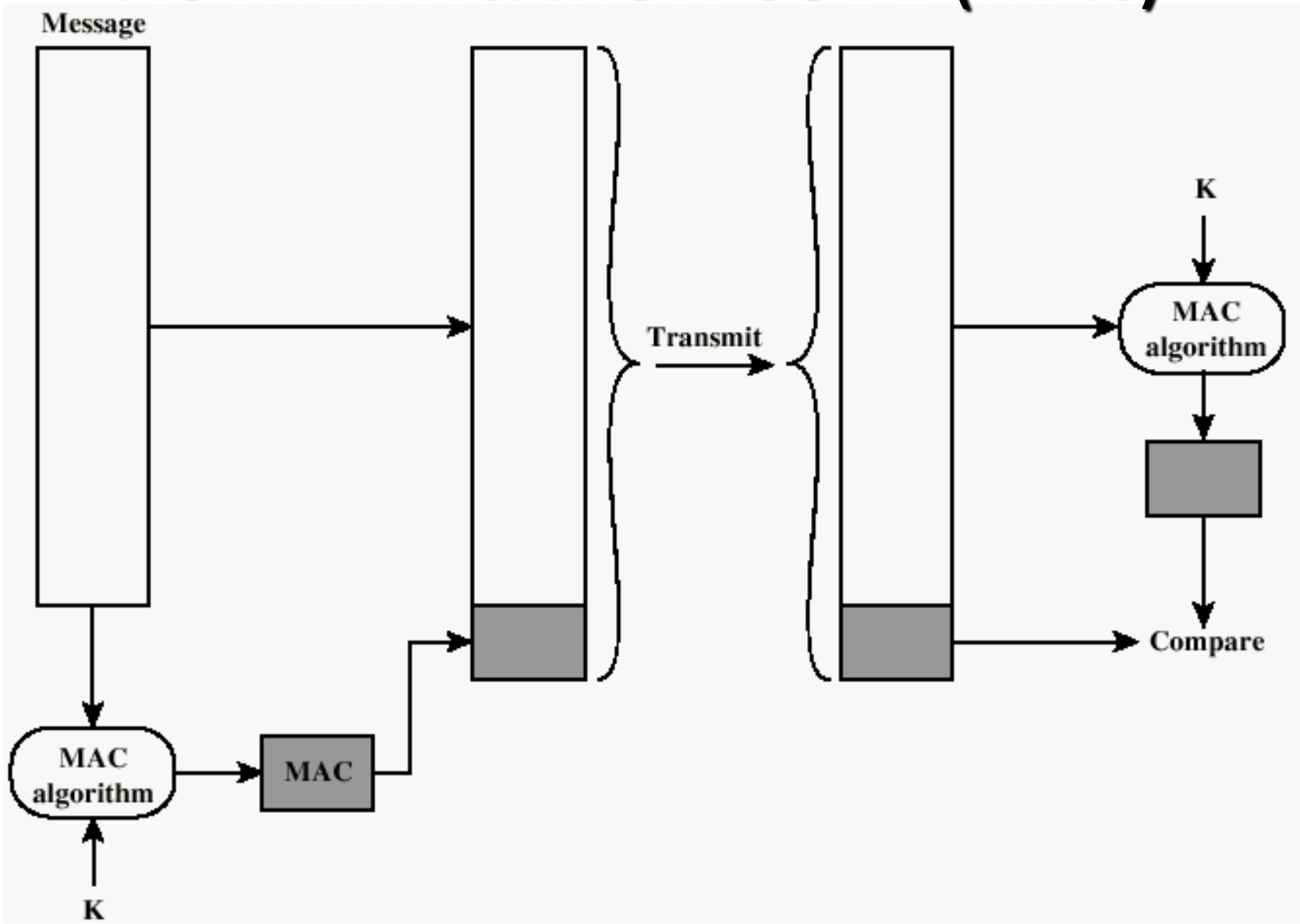
  – αλλοίωση των δεδομένων ή των συναλλαγών

ΜΗΧΑΝΙΣΜΟΙ MESSAGE AUTHENTICATION

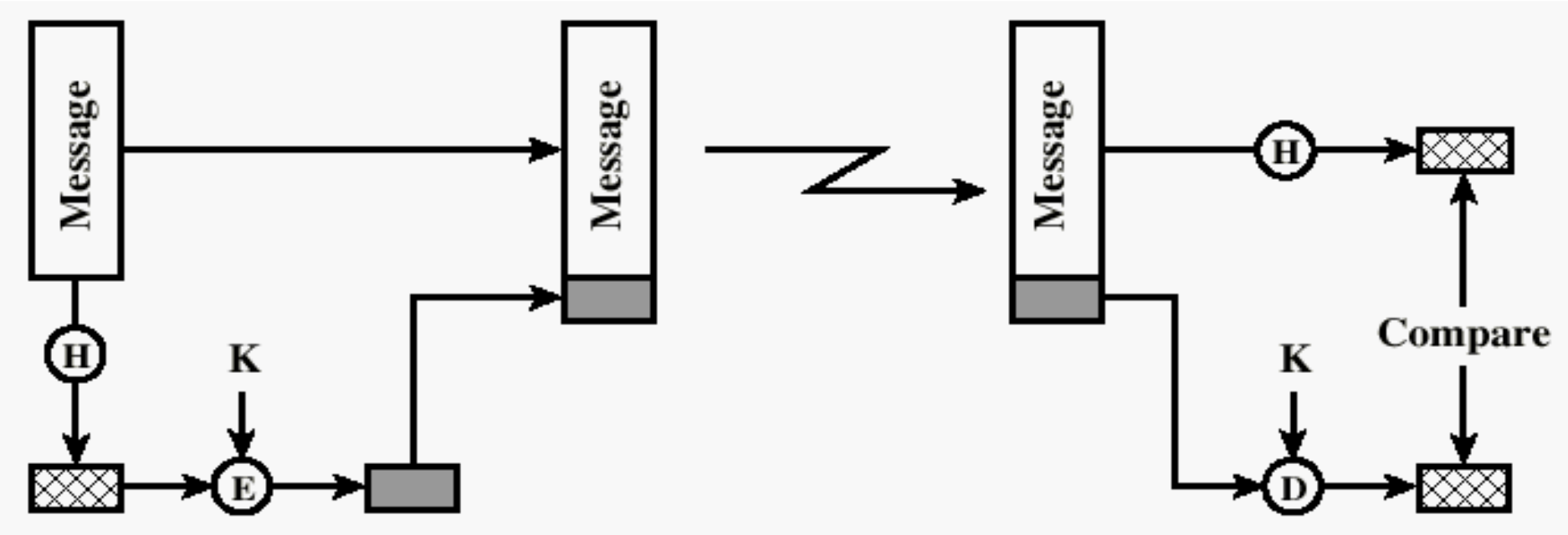
- **Authentication Using Conventional Encryption :**
  - *Μόνο ο αποστολέας και ο παραλήπτης διαμοιράζονται ένα μυστικό κλειδί*
- **Message Authentication without Message Encryption :**
  - *Ένα στοιχείο πιστοποίησης (authentication tag) δημιουργείται και επισυνάπτεται σε κάθε μήνυμα*
- Message Authentication Code (MAC):
  - *Υπολογισμός ενός MAC ως συνάρτηση (F) του μηνύματος (M) και κάποιου κλειδιού (K)*

MAC = Function F (Key, Message)

ΑΡΧΙΤΕΚΤΟΝΙΚΗ MESSAGE AUTHENTICATION CODE (MAC)

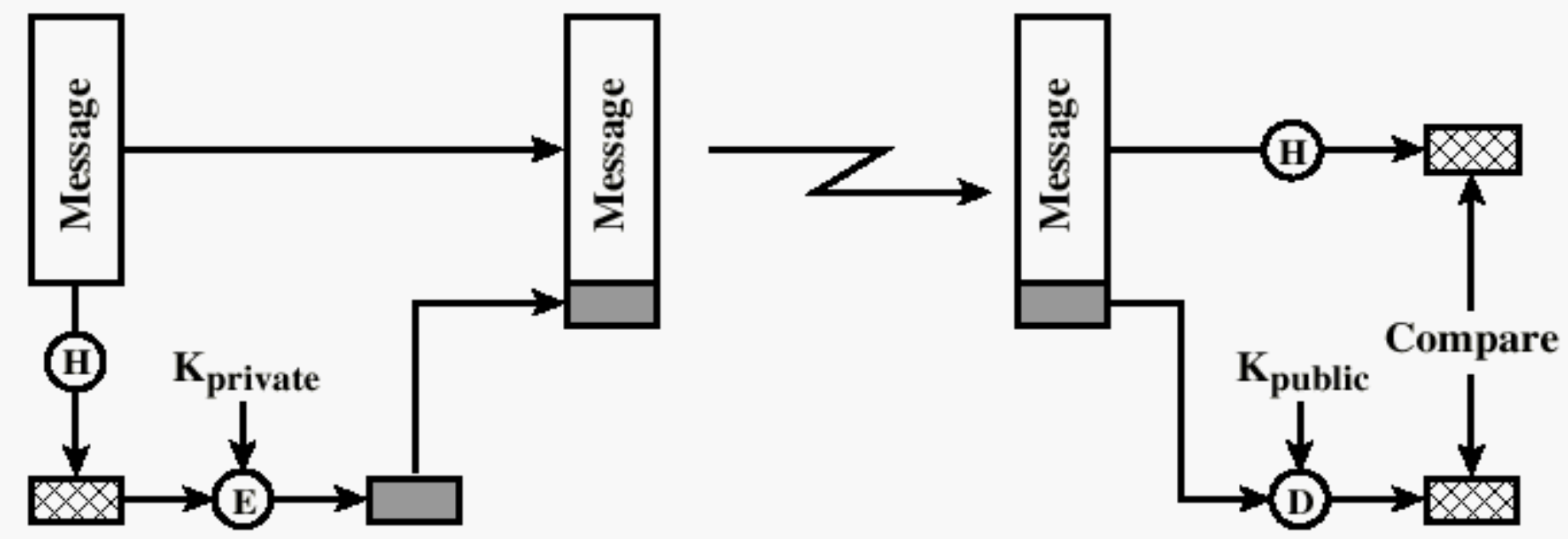


MESSAGE AUTHENTICATION : ΣΥΜΜΕΤΡΙΚΗ ΚΡΥΠΤΟΓΡΑΦΗΣΗ



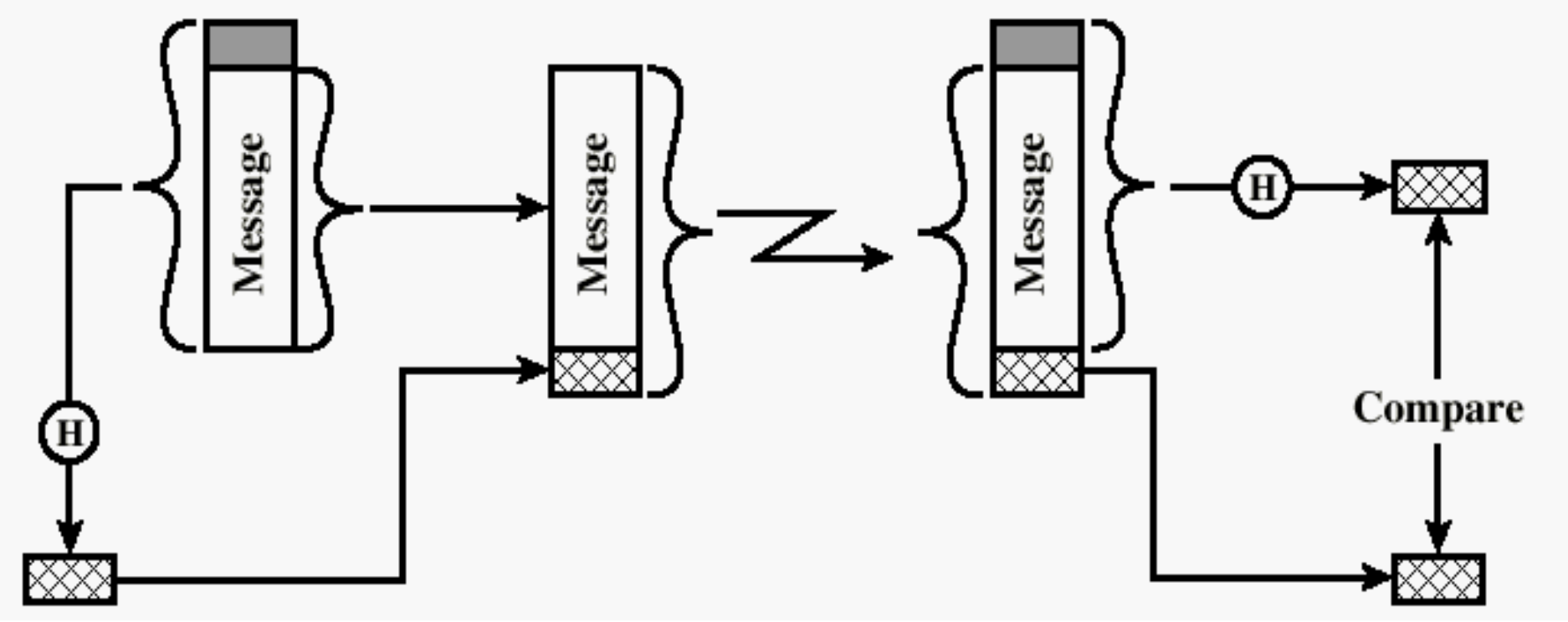
- Χρησιμοποίηση μιας Hash Function
- Συμμετρική Κρυπτογράφηση: Μυστικό Κλειδί K
- Συνένωση του αρχικού μηνύματος (message) με το κρυπτογραφημένο κείμενο

MESSAGE AUTHENTICATION : ΑΣΥΜΜΕΤΡΗ ΚΡΥΠΤΟΓΡΑΦΗΣΗ



- Χρησιμοποίηση μιας Hash Function
- Ασύμμετρη Κρυπτογράφηση: Μυστικό Κλειδί $K_{private}$, Δημόσιο Κλειδί K_{public}
- Συνένωση του αρχικού μηνύματος (message) με το κρυπτογραφημένο κείμενο

MESSAGE AUTHENTICATION: ΧΡΗΣΗ ΜΥΣΤΙΚΗΣ ΤΙΜΗΣ



- Η μυστική πληροφορία επισυνάπτεται πριν τον υπολογισμό hash και αφαιρείται πριν την μετάδοση του μηνύματος
- Δεν απαιτείται η χρήση κλειδιών

HASH MESSAGE AUTHENTICATION CODE (HMAC)

- Τα τελευταία χρόνια παρουσιάζει ιδιαίτερο ενδιαφέρον ο σχεδιασμός ενός MAC με την χρήση hash functions
- Οι λόγοι αυτής της κατεύθυνσης στο σχεδιασμό είναι:
  1. Οι συναρτήσεις τύπου hash function έχουν καλύτερες αποδόσεις σε υλοποιήσεις με την χρήση λογισμικού (software), από ότι οι αλγόριθμοι συμμετρικής κρυπτογράφησης όπως ο DES
  2. Υπάρχουν διαθέσιμοι έτοιμοι κώδικες βιβλιοθηκών (library codes) σε προγραμματίστηκα πακέτα και πλατφόρμες

ΣΚΟΠΟΣ ΣΧΕΔΙΑΣΜΟΥ HMAC

1. Χρησιμοποίηση χωρίς διαφοροποιήσεις μιας συνάρτησης τύπου hash function, που έχει καλύτερες αποδόσεις σε υλοποιήσεις με την χρήση λογισμικού (software). Υπάρχουν διαθέσιμοι έτοιμοι κώδικες βιβλιοθηκών (library codes) σε πακέτα προγραμματισμού & πλατφόρμες

2. Είναι εφικτή η αναβάθμιση και η τροποποίηση του συστήματος με χρήση μιας διαφορετικής hash function, που μπορεί να είναι καλύτερη : i) όσο αφορά το επίπεδο ασφαλείας, ii) την απόδοση υλοποίησης

3. Η δυνατότητα διατήρησης των προδιαγραφών της hash function, χωρίς την υποβάθμιση του τρόπου λειτουργίας της

4. Η χρησιμοποίηση και η διαχείριση κλειδιών με απλό και εφικτό τρόπο

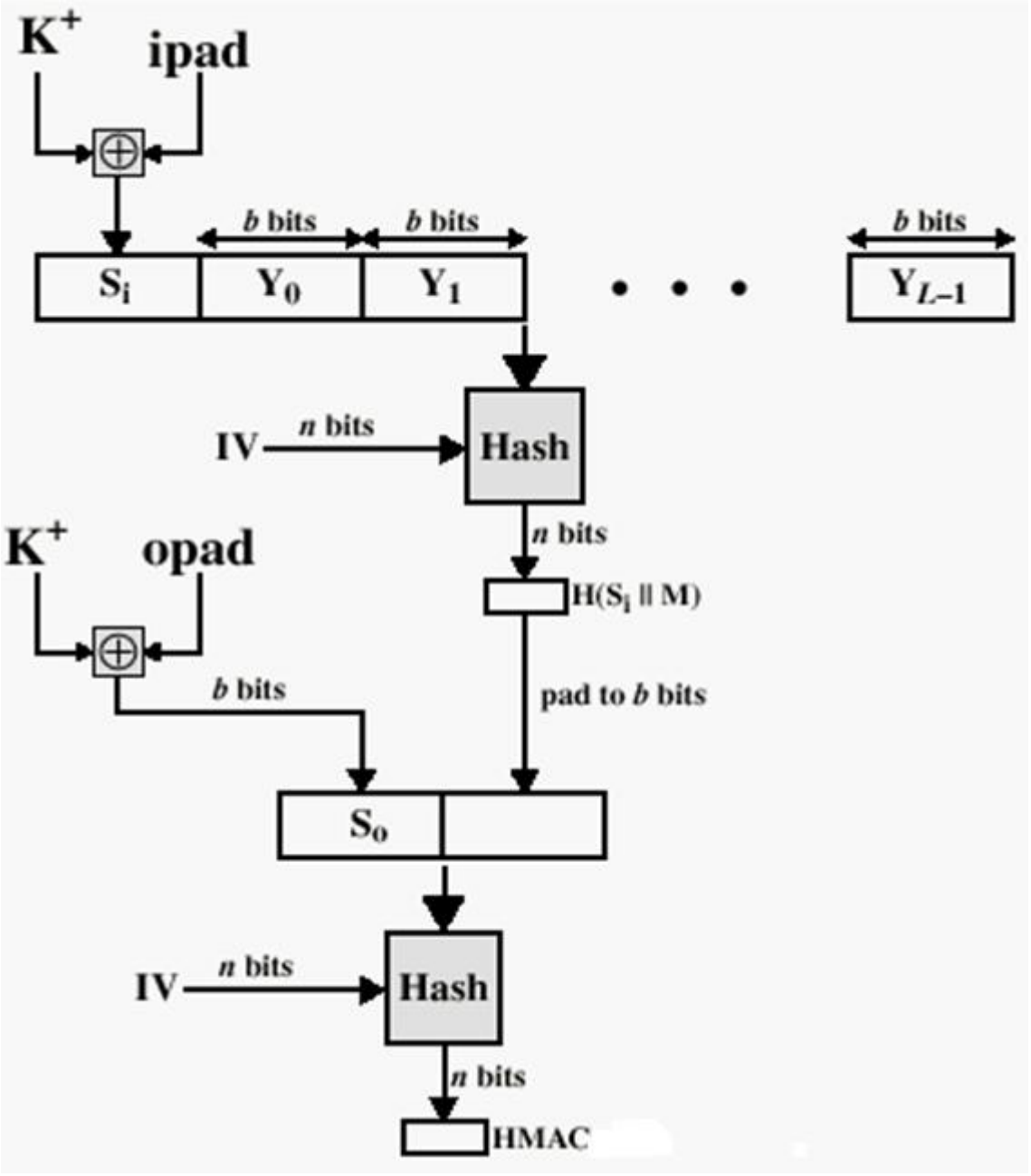
5. Η ανάλυση του επιπέδου ασφαλείας του μηχανισμού πιστοποίησης, στηρίζεται αποκλειστικά και μόνο στις θεωρήσεις ασφαλείας της συνάρτησης hash function

ΑΛΓΟΡΙΘΜΟΣ HMAC

Παράμετροι λειτουργίας του HMAC:

- H : επιλεγμένη Hash Function.
- M : μήνυμα εισόδου του HMAC.
- Yi: i-στο block του μηνύματος εισόδου 0 =< i =< (L-1).
- L: ο μέγιστος αριθμός των block του μηνύματος.
- b: ο αριθμός των ψηφίων σε κάθε block.
- n: το μέγεθος της τιμής εξόδου της Hash Function.
- K: μυστικό κλειδί
- K+: Key padded με μηδενικά.
- ipad: σταθερά 00110110... επαναλαμβανόμενη b/8 φορές
- opad: σταθερά 01011100... επαναλαμβανόμενη b/8 φορές

HMAC ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΛΕΙΤΟΥΡΓΙΑΣ



ECSALab

Απορίες???