Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Εθνικό Μετσόβιο Πολυτεχνείο
Ροή Λ, 9ο εξάμηνο

Υπολογιστική Κρυπτογραφία

Τρίτη σειρά ασκήσεων

Σπουδαστής
Παπασκαρλάτος Αλέξανδρος (Α.Μ.: 03111097)

Ημερομηνία Υποβολής: 9 Δεκεμβρίου 2019

Ο κώδικας που ζητείται βρίσκεται σε ξεχωριστό αρχείο.

Άσκηση 1

Γεννήτρια ψευδοτυχαίων RC4

Αρχικοποίηση

```
for all i ∈ {0..255}
        do : P[i] = i
select K[0...keylen-1]
```

KSA

```
j = 0 for i = 0 to 255 do :
        j = (j + P[i] + K[i mod keylen]) mod 256
        swap(P[i], P[j])
```

PRGA

```
i = 0; j = 0
while next key needed :
        i = (i + 1) mod 256 ; j = (j + P[i]) mod 256
        swap(P[i], P[j])
        K_o = P[(P[i] + P[j]) mod 256]
        output K_o
```

Θα αποδείξουμε ότι το δεύτερο byte εξόδου είναι ίσο με 0 με πιθανότητα περίπου ίση με 2^{-7}.

Καταρχάς, θα δείξουμε ότι αν μετά τη φάση δημιουργίας κλειδιών (KSA) ισχύει για την μετάθεση P ότι P[2] = 0 και P[1] ≠ 2 τότε το δεύτερο byte εξόδου είναι ίσο με 0 με πιθανότητα 1.
Έστω λοιπόν πως μετά τη φάση KSA, έχουμε P[2] = 0 και P[1] ≠ 2.
Στο πρώτο iteration της φάσης παραγωγής ψευδοτυχαίων byte (PRGA), έχουμε:
i := 1, j := 0 + P[i] = P[1], swap(P[1], P[P[1]])
Από αυτά τα βήματα βλέπουμε πως στο τέλος του iteration η θέση j του πίνακα P θα περιέχει το στοιχείο P[1] = j, δηλαδή P[j] = j

Στο δεύτερο iteration της φάσης PRGA, έχουμε:
i := 2, j := j + P[2] = j + 0 = j , καθώς αρχικά P[2] = 0 και επειδή P[1] ≠ 2, δεν έγινε swap του P[2] στο πρώτο iteration, οπότε δεν πειράχτηκε το P[2]. Οπότε, το j παραμένει σταθερό.
swap(P[2], P[j]) , άρα, σύμφωνα με τα προηγούμενα, προκύπτει P[j] = 0, P[2] = j
K_o := P[(P[i] + P[j])] = P[P[2] + P[j]] = P[j + 0] = P[j] = 0

Επομένως, πράγματι, αν μετά τη φάση KSA ισχύει P[2] = 0 και P[1] ≠ 2 τότε το δεύτερο byte εξόδου είναι ίσο με 0.

Ας δείξουμε τώρα πότε το P[2] καταλήγει να είναι 0 μετά τη φάση KSA. Κατά τη φάση KSA, ο πίνακας P προκύπτει από μετάθεση των [0...255]. Για κλειδί μήκους 256 επιλεγμένο τυχαίο από ομοιόμορφη κατανομή, προκύπτει ομοιόμορφα τυχαία μετάθεση των [0...255]. Συνεπώς, το P[2] καταλήγει να είναι 0 με πιθανότητα 1/256 = 2^{-8}. Εφόσον θέλουμε επιπλέον και P[1] ≠ 2, η ζητούμενη πιθανότητα είναι λίγο μικρότερη, $2^{-8} \cdot (254/255)$, δηλαδή περίπου 2^{-8}.

Επίσης, ακόμα και όταν P[2] ≠ 0 ∨ P[1] = 2, το δεύτερο byte της εξόδου έχει μία πιθανότητα να είναι ίσο με 0 συμπτωματικά. Επειδή το byte αποτελείται από 8 bit, εξετάζουμε την περίπτωση που και τα 8 bit είναι 0. Επειδή η RC4 είναι PRG, όταν δεν ισχύει πως P[2] = 0 και P[1] ≠ 2, μπορούμε να θεωρήσουμε πως το byte αυτό προκύπτει ομοιόμορφα τυχαία, άρα με πιθανότητα 2^{-8} (αφού υπάρχουν 2^8 συνδυασμοί των 8 δυαδικών bit). Η πιθανότητα αυτή ίσως είναι ανακριβής, ειδικά εάν η RC4 έχει και άλλες “κακές” ιδιότητες, ιδιότητες που την καθιστούν κακή PRG, αλλά αυτό μάλλον ξεφεύγει από τα ζητούμενα της άσκησης και το 2^{-8} είναι ό,τι καλύτερο γνωρίζουμε με τα στοιχεία που έχουμε.

Επομένως, συνολικά, η πιθανότητα το δεύτερο byte της εξόδου να ισούται με 0 είναι περίπου $2^{-8} + 2^{-8} = 2^{-7}$.

Άσκηση 2

Υλοποίησα τη γεννήτρια ψευδοτυχαίων bit BBS σε c++.
Ο κώδικας βρίσκεται στο αρχείο Crypto_ex_3_2.cpp .
Στο αρχείο output.txt φαίνονται κάποια αποτελέσματα (η περίοδος, κάποια στοιχεία s_i, οι επιτυχίες, οι προσεγγίσεις, κλπ) .

a. Για μια οποιαδήποτε γεννήτρια BBS, δηλαδή για οποιαδήποτε επιλογή πρώτων p, q, με $p \equiv q \equiv 3 \pmod 4$, ισχύουν τα κάτωθι:

- Έστω T η περίοδος της γεννήτριας, s ο αρχικός σπόρος και $n = p \cdot q$.
- Τότε υπάρχει i, τέτοιο ώστε
- $s^{2\wedge(i+T)} \equiv s^{2\wedge i} \pmod n \quad \Rightarrow \quad 2^{i+T} \equiv 2^i \pmod r$, όπου $r = ord_n(s)$
- Αν το r δεν είναι σχετικά πρώτο με το 2^{i+T}, δηλαδή το r είναι πολλαπλάσιο κάποιας δύναμης του 2, έστω 2^k , τότε και το δεξί μέλος της ισοτιμίας πρέπει να είναι πολλαπλάσιο της ίδιας δύναμης του 2, προκειμένου η ισοτιμία να έχει λύση.
  Δηλαδή πρέπει $k \geq i$, οπότε και προκύπτει :
  $2^{i+T} / 2^k \equiv 2^i / 2^k \pmod{r/2^k} \quad \Rightarrow \quad 2^{i-k+T} \equiv 2^{i-k} \pmod{r/2^k} \quad \Rightarrow$
  $2^{i-k+T} / 2^{i-k} \equiv 2^{i-k} / 2^{i-k} \pmod{r/2^k} \quad \Rightarrow \quad 2^T \equiv 1 \pmod{r/2^k}$
- Συνεπώς, $\mathbf{T \mid \lambda(r / 2^k)}$,
  όπου 2^k η μεγαλύτερη δύναμη του 2 που είναι υποπολλαπλάσιο του $r = ord_n(s)$
  και λ() είναι η συνάρτηση Carmichael.
- Μία πιο γενική σχέση, χωρίς να μπλέξουμε με τάξεις στοιχείων, είναι η κάτωθι:
  $r/2^i \mid r \mid \lambda(n) \quad \Rightarrow \quad \lambda(r/2^i) \mid \lambda(\lambda(n)) \quad \Rightarrow \quad \mathbf{T \mid \lambda(\lambda(n))}$

  Η σχέση αυτή παρέχει λιγότερη πληροφορία από την προηγούμενη,
  καθώς $\lambda(r/2^i) \leq \lambda(\lambda(n))$, αλλά αξιοποιεί μόνο το n (δηλαδή μόνο τα p, q).
  Συνεπώς, είναι πιο γενική πληροφορία για τη γεννήτρια και ισχύει για οποιαδήποτε επιλογή αρχικού σπόρου.

Είναι $\lambda(n) = \lambda(p \cdot q) = lcm(p-1, q-1)$
Για $d = gcd(p-1, q-1)$, έχουμε :

- $\exists$ k, m: $\quad p-1 = k \cdot d \quad \wedge \quad q-1 = m \cdot d$
- $gcd((p-1)/d , (q-1)/d) = gcd(k,m) = 1$
- $lcm(p-1, q-1) = d \cdot k \cdot m = (p-1) \cdot (q-1) / gcd(p-1, q-1)$
- $\lambda(n) = \lambda(p \cdot q) = lcm(p-1, q-1) = (p-1) \cdot (q-1) / gcd(p-1, q-1)$

Οπότε, για μικρό gcd, έχουμε μεγαλύτερο λ(n), άρα και ίσως μεγαλύτερη περίοδο, σύμφωνα με τις σχέσεις που δείξαμε νωρίτερα.

Είναι $p \equiv 3 \pmod 4 \Rightarrow \quad \exists\ k : p = 4k + 3 = 2 \cdot (2k) + 2 + 1 = 2 \cdot (2k + 1) + 1 \Rightarrow$
για $p' = 2k + 1$: $\quad p = 2p' + 1$
Όμοια, $\exists\ q'$: $\quad q = 2q' + 1$

Εάν φροντίσω ώστε τα p', q' είναι πρώτοι, δηλαδή εάν έχω διαλέξει p, q safe primes, προκύπτει πως gcd(p-1, q-1) = gcd(2p', 2q') = 2 , το οποίο είναι το μικρότερο δυνατό gcd.

Για τη δική μου υλοποίηση, επέλεξα:
safe primes p = 600239, q = 600983, άρα n = 360733434937
seed = $20749^2 \equiv 430521001 \pmod n$

Ενδεικτικά τα πρώτα 100 bit εξόδου της γεννήτριάς μου είναι:
1011001110001110010010010111011100001000011100100111011000111111110001011011
0100100000000100000110111

Η περίοδος προέκυψε πειραματικά T = 45091228910 = λ(λ(n))

b. i. Καταρχάς, αναφέρουμε πως το κόλπο είναι στημένο και παρά τα λεγόμενα της εκφώνησης, δεν μπορούμε ποτέ να προσεγγίσουμε το π με καλή ακρίβεια με αυτή τη μέθοδο.
Το πρόβλημα έγκειται στο γεγονός πως το τετράγωνο και ο κύκλος που κατασκευάζουμε έχουν πολύ λίγα σημεία.
Στην πράξη, ο κύκλος περιέχει 51.040 σημεία από τα συνολικά 65.536 σημεία του τετραγώνου.
Συνεπώς, εάν τα σημεία επιλέγονταν τυχαία με ομοιόμορφη κατανομή, το κάθε επιλεγόμενο σημείο θα είχε πιθανότητα 51040 / 65536 = 0,77880859375 = 3,115234375 / 4 να ανήκει στον κύκλο.
Επομένως, η καλύτερη δυνατή θεωρητική προσέγγιση που θα μπορούσαμε να πετύχουμε για το π, είναι π' = 3,115234375.

Για το υπόλοιπο του ερωτήματος, θα ασχοληθούμε με το π', όχι το π.

<u>Θεωρητική προσέγγιση</u>
Κάθε bit εξόδου είναι ίσο με '1' με πιθανότητα ½ .
Προκύπτει πως κάθε σημείο είναι ισοπίθανο να επιλεγεί με πιθανότητα $1/65536 = 2^{-16}$.
Κάθε τέτοια επιλογή αποτελεί ένα πείραμα Bernoulli με πιθανότητα επιτυχίας
p = 0,77880859375 .
Προκειμένου να πετύχουμε απόκλιση το πολύ d στο π' = 4π, χρειαζόμαστε απόκλιση το πολύ d/4 στο p.

Από Chernoff bound για διαδοχικές δοκιμές Bernoulli, ισχύουν οι σχέσεις:

- $Pr(X \le (1 - \delta) \mu) \le e^{- \delta \delta \mu / 2}$
- $Pr(X \ge (1+ \delta) \mu) \le e^{- \delta \delta \mu / (2 + \delta)}$

όπου X το άθροισμα των αποτελεσμάτων των n δοκιμών, $\mu = n \cdot p$ η αναμενόμενη τιμή και
$0 \le \delta \le 1$ αυθαίρετα επιλεγμένο.

Επομένως,

- $Pr(X/n \le p - \delta \cdot p) \le e^{- \delta \delta \mu / 2} \le e^{- \delta \delta \mu / (2 + \delta)}$
- $Pr(X/n \ge p + \delta \cdot p) \le e^{- \delta \delta \mu / (2 + \delta)}$

όπου $\delta \cdot p$ το όριο της αποδεκτής απόκλισης.
Το μέσο X που προκύπτει πειραματικά θα αποκλίνει από την αναμενόμενη τιμή κατά ποσότητα μεγαλύτερη της αποδεκτής απόκλισης με πιθανότητα μικρότερη της ποσότητας $e^{- \delta \delta \mu / (2 + \delta)}$.

Θα θεωρήσουμε πως θέλουμε να πετυχαίνουμε την εκάστοτε ακρίβεια με πιθανότητα περίπου 99%, άρα να ξεφεύγουμε από την αποδεκτή απόκλιση με πιθανότητα μικρότερη ή ίση του 1%. Σημειώνουμε πως μια τέτοια πιθανότητα για μια τόσο μικρή απόκλιση είναι αρκετά “απαιτητική”.

<u>ακρίβεια 2 δεκαδικών ψηφίων</u>

$d = 0,005 \quad \Rightarrow \quad d/4 = 0,00125$

$\delta \cdot p = d / 4 \quad \Rightarrow \quad \delta = 0,001605$

Θέλουμε $e^{- \delta \delta \mu / (2 + \delta)} \le 0,01 \quad \Rightarrow \quad - \delta^2 n p / (2+\delta) \le \ln(0,01) \quad \Rightarrow$

$n \ge - \ln(0,01) \cdot (2+\delta) / (\delta^2 p) \quad \Rightarrow$

$n \ge 4594552,866250$

Επομένως, χρειαζόμαστε τουλάχιστον 4.594.553 σημεία.

<u>ακρίβεια 3 δεκαδικών ψηφίων</u>

$d = 0,0005 \quad \Rightarrow \quad d/4 = 0,000125$

$\delta \cdot p = d / 4 \quad \Rightarrow \quad \delta = 0,0001605$

Θέλουμε $e^{- \delta \delta \mu / (2 + \delta)} \le 0,01 \quad \Rightarrow \quad n \ge 459123711,13359447$

Επομένως, χρειαζόμαστε τουλάχιστον 459.123.712 σημεία.

<u>ακρίβεια 4 δεκαδικών ψηφίων</u>

$d = 0,0005 \quad \Rightarrow \quad d/4 = 0,0000125$

$\delta \cdot p = d / 4 \quad \Rightarrow \quad \delta = 0,00001605$

Θέλουμε $e^{- \delta \delta \mu / (2 + \delta)} \le 0,01 \quad \Rightarrow \quad n \ge 45909055358,445116597$

Επομένως, χρειαζόμαστε τουλάχιστον 45.909.055.358 σημεία.

<u>Πειραματική προσέγγιση</u>

Μέσω της υλοποίησής μας, παρατηρούμε από ποιο πλήθος σημείων και μετά πετυχαίνουμε την εκάστοτε ακρίβεια.

Σημειώνουμε πως δεν αρκεί να πετύχουμε μία αποδεκτή προσέγγιση για κάποιο συγκεκριμένο πλήθος σημείων. Απαιτούμε να διατηρείται αυτή η προσέγγιση εντός ορίων καθώς αυξάνεται το πλήθος των σημείων από εκεί και έπειτα.

<u>ακρίβεια 2 δεκαδικών ψηφίων</u>

Χρειαζόμαστε περίπου 450.000 bit εξόδου, άρα 28.125 σημεία.

Ας κάνουμε και εδώ μια σύγκριση με θεωρητικά αποτελέσματα.

Για X = #successes = 21881 (πειραματικά) στα n = 28125 σημεία.

και $\mu = n \cdot p = 28125 \cdot 0,77880859375 = 21903,9916978125$

και $X = (1 - \delta) \cdot \mu \Rightarrow \delta = (\mu - X) / \mu = 0,00104965789$

$\Rightarrow e^{-\delta \delta \mu / 2} = 0,988005802$

Για επιλογή n πραγματικά τυχαίων σημείων, έχουμε:

- $Pr(X \geq (1- \delta) \mu) \leq e^{- \delta \delta \mu / 2}$

Επομένως, εάν παίρναμε n πραγματικά τυχαία σημεία, το αποτέλεσμα των επιτυχιών θα απείχε περισσότερο από την αναμενόμενη τιμή από το αποτέλεσμα που προέκυψε πειραματικά με πιθανότητα P μικρότερη από 0,988005802.

Φυσικά για να εξάγουμε ισχυρό τυπικό συμπέρασμα, θα επιθυμούσαμε να έχουμε κάποιο κάτω όριο για την πιθανότητα αυτή, αντί για άνω όριο.

Ωστόσο, ακόμα και αυτό το άνω όριο μας δίνει κάποια ιδέα για το αποτέλεσμα της τυχαίας επιλογής, καθώς κατά πιθανότητα P, η τυχαία επιλογή θα έδινε αποτελέσματα που απέχουν περισσότερο από την αναμενόμενη τιμή, οπότε θα έμοιαζε “λιγότερο τυχαία”.

<u>ακρίβεια 3 δεκαδικών ψηφίων</u>

Χρειαζόμαστε περίπου 310.000.000 bit εξόδου, άρα 19.375.000 σημεία.

Ας κάνουμε και εδώ μια σύγκριση με θεωρητικά αποτελέσματα.

Για X = #successes = 15091863 (πειραματικά) στα n = 19375000 σημεία,

και $\mu = n \cdot p = 19375000 \cdot 0,77880859375 = 15089416,50390625$

και $X = (1 + \delta) \cdot \mu \Rightarrow \delta = (X - \mu) / \mu = 0,0001613324$

$\Rightarrow e^{-\delta \delta \mu / (2+\delta)} = 0,82171726$

Για επιλογή n πραγματικά τυχαίων σημείων, έχουμε:

- $Pr(X \geq (1+ \delta) \mu) \leq e^{- \delta \delta \mu / (2 + \delta)}$

Επομένως, εάν πέρναμε n πραγματικά τυχαία σημεία, το αποτέλεσμα των επιτυχιών θα απείχε περισσότερο από την αναμενόμενη τιμή από το αποτέλεσμα που προέκυψε πειραματικά με πιθανότητα P μικρότερη από 0,82171726.

Φυσικά για να εξάγουμε ισχυρό τυπικό συμπέρασμα, θα επιθυμούσαμε να έχουμε κάποιο κάτω όριο για την πιθανότητα αυτή, αντί για άνω όριο.
Ωστόσο, ακόμα και αυτό το άνω όριο μας δίνει κάποια ιδέα για το αποτέλεσμα της τυχαίας επιλογής, καθώς κατά πιθανότητα P, η τυχαία επιλογή θα έδινε αποτελέσματα που απέχουν περισσότερο από την αναμενόμενη τιμή, οπότε θα έμοιαζε “λιγότερο τυχαία”.

ακρίβεια 4 δεκαδικών ψηφίων
Χρειαζόμαστε περίπου 3.920.000.000 bit εξόδου, άρα 245.000.000 σημεία.

Ας κάνουμε και εδώ μια σύγκριση με θεωρητικά αποτελέσματα.
Για $X = \#successes = 190804631$ (πειραματικά) στα $n = 245000000$ σημεία,
και $\mu = n \cdot p = 245000000 \cdot 0{,}77880859375 = 190808105{,}46875$
και $X = (1 - \delta) \cdot \mu \quad \Rightarrow \quad \delta = (\mu - X) / \mu = 0{,}00001820923$
$\Rightarrow \quad e^{-\delta \delta \mu / 2} = 0{,}968861411$
Για επιλογή n πραγματικά τυχαίων σημείων, έχουμε:

- $Pr(X \geq (1- \delta) \mu) \leq e^{- \delta \delta \mu / 2}$

Επομένως, εάν παίρναμε n πραγματικά τυχαία σημεία, το αποτέλεσμα των επιτυχιών θα απείχε περισσότερο από την αναμενόμενη τιμή από το αποτέλεσμα που προέκυψε πειραματικά με πιθανότητα P μικρότερη από 0,968861411.
Φυσικά για να εξάγουμε ισχυρό τυπικό συμπέρασμα, θα επιθυμούσαμε να έχουμε κάποιο κάτω όριο για την πιθανότητα αυτή, αντί για άνω όριο.
Ωστόσο, ακόμα και αυτό το άνω όριο μας δίνει κάποια ιδέα για το αποτέλεσμα της τυχαίας επιλογής, καθώς κατά πιθανότητα P, η τυχαία επιλογή θα έδινε αποτελέσματα που απέχουν περισσότερο από την αναμενόμενη τιμή, οπότε θα έμοιαζε “λιγότερο τυχαία”.

b. ii. Με μια πολύ μικρή αλλαγή στον κώδικα, δημιουργούμε παραλλαγή της BBS, όπου τα bit εξόδου της γεννήτριας είναι η ισοτιμία των δυαδικών ψηφίων των αριθμών s_i .

Ενδεικτικά τα πρώτα 100 bit εξόδου είναι:
1111100110000001110010100001010001101000000111010101011101001110101100101110
0010110110111111101111101

Παρακάτω παρουσιάζουμε πόσα bit εξόδου χρειαζόμαστε για να πετύχουμε ακρίβεια 2 / 3 / 4 δεκαδικών ψηφίων στο π’ .

ακρίβεια 2 δεκαδικών ψηφίων
Χρειαζόμαστε περίπου 980.000 bit εξόδου, άρα 61.250 σημεία.

ακρίβεια 3 δεκαδικών ψηφίων
Χρειαζόμαστε περίπου 130.000.000 bit εξόδου, άρα 8.125.000 σημεία.

ακρίβεια 4 δεκαδικών ψηφίων
Χρειαζόμαστε περίπου 24.360.000.000 bit εξόδου, άρα 1.522.500.000 σημεία.

Συγκρίνοντας, με τα αποτελέσματα που λάβαμε από την απλή BBS, βλέπουμε πως για να πετύχουμε ακρίβεια 2 δεκαδικών ψηφίων χρειαζόμαστε περίπου τα διπλάσια σημεία, για 3 δεκαδικών, περίπου τα μισά, ενώ για 4 δεκαδικών, περίπου τα εξαπλάσια.

Παρατηρούμε πως χρειαζόμαστε κάθε φορά την ίδια τάξη μεγέθους πλήθους σημείων με πριν. Παράλληλα, παρατηρούμε πως δε χρειαζόμαστε σταθερά περισσότερα ή λιγότερα σημεία για την εκάστοτε ακρίβεια (πχ για ακρίβεια 2 ψηφίων χρειαζόμαστε περισσότερα σημεία, ενώ για ακρίβεια 3, χρειαζόμαστε λιγότερα). Αντ' αυτού οι τιμές κυμαίνονται εντός πιθανοτικών ορίων. Οι διαφορές στα αποτελέσματα των δύο εκδοχών της γεννήτριας μοιάζουν τυχαίες και συμπτωματικές, κάτι που είναι δόκιμο σε γεννήτριες ψευδοτυχαιότητας.

c. Καταρχάς αναφέρουμε πως το ερώτημα (β) θα μπορούσε δικαίως να θεωρηθεί ως ένα τεστ ψευδοτυχαιότητας, καθώς υπάρχει μια σύγκριση ανάμεσα στα bit εξόδου μιας ψευδοτυχαίας γεννήτριας και της θεωρητικής τυχαίας τιμής από ομοιόμορφη κατανομή.

Επιπλέον επιτελούμε το Frequency (Monobit) Test.
Ελέγχουμε το συνολικό πλήθων των άσσων που προκύπτουν σε μία περίοδο της γεννήτριας προς το συνολικό πλήθος των bit.
Φυσικά, αν επιλέγονταν τυχαία bit με ομοιόμορφη κατανομή, ο λόγος αυτός θα έτεινε στο ½ .
Τα πειραματικά αποτελέσματα είναι :
Απλή BBS (bit εξόδου i : s_i (mod 2)) :
22545699312 / 45091228910 = 0,50000188189592635345187357414163
Παραλλαγή BBS (bit εξόδου i : ισοτιμία των bits του s_i) :
22545497580 / 45091228910 = 0,49999740803249711208636030053145

Παρατηρούμε πως και οι δύο εκδοχές της γεννήτριας παρουσιάζουν πολύ μικρή απόκλιση από το θεωρητικό μέσο κατά την ομοιόμορφα τυχαία επιλογή, δηλαδή περνάνε το τεστ.

Άσκηση 3

Θεωρούμε την παραλλαγή του DES-X, όπου
$Enc_{k1, k2}(M) = E_{k1}(M \oplus k_2)$, όπου E η συνάρτηση κρυπτογράφησης του DES.

Θεωρούμε πως ο αντίπαλος έχει δυνατότητα KPA.

Θα αποδείξουμε πως το σύστημα αυτό δεν βελτιώνει ουσιαστικά την ασφάλεια του απλού DES.

Καταρχάς, θυμίζουμε τα εξής για την πράξη $\oplus$ (XOR) :

$\forall$ x. $x \oplus 0 = 0 \oplus x = x$, οπότε το 0 είναι ουδέτερο στοιχείο της $\oplus$

και

$\forall$ x. $x \oplus x = 0$

Τα παραπάνω ισχύουν για ένα bit, και επαγωγικά ισχύουν για ακολουθίες από bit.
Για μήκος n, φυσικά εννοούμε πως το 0^n είναι το ουδέτερο στοιχείο και $x \oplus x = 0^n$.

Έστω πως είμαστε ο αντίπαλος. Θα ακολουθήσουμε τα εξής βήματα.

- Έχουμε ζεύγη κειμένων, κρυπτοκειμένων: (M_1, C_1), (M_2, C_2).
- Θεωρούμε τις ποσότητες $T_1 = M_1 \oplus k_2$ και $T_2 = M_2 \oplus k_2$
  Σημειώνουμε πως, για την ώρα, το k_2 είναι άγνωστο, οπότε τα T_1, T_2 είναι άγνωστα.
- Θεωρούμε την ποσότητα $T = T_1 \oplus T_2 = M_1 \oplus k_2 \oplus M_2 \oplus k_2 = M_1 \oplus M_2$
  Σημειώνουμε πως το T είναι γνωστό, ή πιο σωστά, εύκολα υπολογίσιμο.
- Είναι $C_1 = E_{k1}(T_1)$, $C_2 = E_{k1}(T_2)$.
- Υποθέτουμε κάποια τιμή για το k_1 και δοκιμάζουμε την αποκρυπτογράφηση DES για τα C_1 , C_2 , οπότε προκύπτουν T_1', T_2' .
- Αν $T_1' \oplus T_2' \neq T$, τότε υποθέσαμε λάθος τιμή για το k_1. Ξαναδοκιμάζουμε με άλλη τιμή.
- Αν $T_1' \oplus T_2' = T$, τότε βρήκαμε σωστή τιμή για το k_1 και, επιπλέον, ισχύει $T_1 = T_1'$, $T_2 = T_2'$
  (Σημειώνουμε πως τυπικά, υπάρχει μια πολύ μικρή πιθανότητα να ισχύει $T_1' \oplus T_2' = T$ και να μην έχουμε βρει το σωστό κλειδί k_1 , δηλαδή να πρόκειται για απίθανη σύμπτωση. Η πιθανότητα αυτή είναι μάλλον αμελητέα, αλλά αν θέλουμε να την εξαλείψουμε, απλά δοκιμάζουμε το κλειδί και σε ένα τρίτο γνωστό κρυπτοκείμενο C_3 και εξετάζουμε αν $T_3' \oplus T_1' = T_3 \oplus T_1$ και $T_3' \oplus T_2' = T_3 \oplus T_2$)
- Έχοντας βρει το T_1, είναι $T_1 = M_1 \oplus k_2$ $\Rightarrow$ $M_1 \oplus T_1 = k_2$
  οπότε υπολογίζουμε και το k_2 .

Συνεπώς, σπάσαμε την κρυπτογράφηση.

Σημειώνουμε πως υποθέσαμε πως κάνουμε brute force επίθεση για το κλειδί k_1 του DES.
Ωστόσο, στον κλασικό DES υπάρχουν πιο έξυπνες και αποδοτικές επιθέσεις KPA.
Ομολογώ πως δε γνωρίζω αν μπορούν να εφαρμοστούν εδώ εξυπνότερες επιθέσεις, αν και το γεγονός πως το T_1 είναι αρχικά ουσιαστικά τελείως άγνωστο, καθώς ανάλογα με την επιλογή του k_2 μπορεί να προκύψει οποιοδήποτε κείμενο, μάλλον τέτοιες επιθέσεις περιπλέκονται πολύ.
Οπότε, από αυτήν την οπτική, ίσως πράγματι να βελτιώνεται λίγο η ασφάλεια του συστήματος.

Άσκηση 4

Καταρχάς, θυμίζουμε τα εξής για την πράξη $\oplus$ (XOR) :

$\forall$ x. $x \oplus 0 = 0 \oplus x = x$, οπότε το 0 είναι ουδέτερο στοιχείο της $\oplus$

και

$\forall$ x. $x \oplus x = 0$

Τα παραπάνω ισχύουν για ένα bit, και επαγωγικά ισχύουν για ακολουθίες από bit.
Για μήκος n, φυσικά εννοούμε πως το 0^n είναι το ουδέτερο στοιχείο και $x \oplus x = 0^n$.

1. $h_1(x_1||x_2||x_3||x_4) = h(x_1 \oplus h(x_2||x_2))||(h(x_3||x_3) \oplus x_4))$

Η h_1 δεν είναι collision resistant.

Πράγματι, θεωρώ string
μήκους 4n, τυχαίο $X = x_1||x_2||x_3||x_4$,
μήκους n, τυχαίο y_2 , με $y_2 \neq x_2$,
μήκους n, $y_1 = x_1 \oplus h(x_2||x_2) \oplus h(y_2||y_2)$,
μήκους 4n, $Y = y_1||y_2||x_3||x_4$

Σημειώνω πως το y_1 είναι εύκολα υπολογίσιμο για δοθέντα x_1, x_2, y_2 .
Ισχύει $y_1 = x_1 \oplus h(x_2||x_2) \oplus h(y_2||y_2) \Leftrightarrow y_1 \oplus h(y_2||y_2) = x_1 \oplus h(x_2||x_2)$ (1)

Είναι $h_1(X) = h_1(x_1||x_2||x_3||x_4) = h(x_1 \oplus h(x_2||x_2))||(h(x_3||x_3) \oplus x_4))$

$h_1(Y) = h_1(y_1||y_2||x_3||x_4) = h(y_1 \oplus h(y_2||y_2))||(h(x_3||x_3) \oplus x_4))$
$= h(x_1 \oplus h(x_2||x_2))||(h(x_3||x_3) \oplus x_4)) = h_1(X)$

Και επειδή, από κατασκευή, $x_2 \neq y_2$, σίγουρα $X \neq Y$.
Επομένως, βρήκαμε εύκολα σύγκρουση για την h_1 .

2. $h_2(x_1||x_2||x_3||x_4) = h(h(x_1||x_2)||h(x_3||x_4))$

Η h_2 είναι collision resistant.

Block άτοπο: Έστω η h_2 δεν είναι collision resistant

- Μπορούμε να βρούμε εύκολα σύγκρουση για την h_2 , δηλαδή μπορούμε να βρούμε X, Y μήκους 4n με $X \neq Y$ και $h_2(X) = h_2(Y)$.
- Θεωρώ $X = x_1||x_2||x_3||x_4$, $h_2(X) = h(h(x_1||x_2)||h(x_3||x_4))$
- Θεωρώ $Y = y_1||y_2||y_3||y_4$, $h_2(Y) = h(h(y_1||y_2)||h(y_3||y_4))$
- $h(X) = h(Y) \Rightarrow h(h(x_1||x_2)||h(x_3||x_4)) = h(h(y_1||y_2)||h(y_3||y_4))$
- Subblock άτοπο: Έστω $h(x_1||x_2)||h(x_3||x_4) \neq h(y_1||y_2)||h(y_3||y_4)$
  - Θεωρώ $m_1 = h(x_1||x_2)||h(x_3||x_4)$, $m_2 = h(y_1||y_2)||h(y_3||y_4)$
  - Βρήκαμε m_1 , m_2 με $m_1 \neq m_2$ και $h(m_1) = h(m_2)$, δηλαδή βρήκαμε σύγκρουση για την h.

  Άτοπο, καθώς η h είναι collision resistant.
  Επομένως, $h(x_1||x_2)||h(x_3||x_4) = h(y_1||y_2)||h(y_3||y_4)$
- Προκύπτει, $h(x_1||x_2) = h(y_1||y_2) \wedge h(x_3||x_4) = h(y_3||y_4)$
  Όμοια με το προηγούμενο subblock άτοπο, δείχνουμε (ξεχωριστά για την κάθε ισότητα) πως $x_1||x_2 = y_1||y_2 \wedge x_3||x_4 = y_3||y_4$
- Άρα $x_1 = y_1 \wedge x_2 = y_2 \wedge x_3 = y_3 \wedge x_4 = y_4 \Rightarrow X = Y$

Άτοπο, καθώς, από κατασκευή $X \neq Y$.
Επομένως, η h_2 είναι collision resistant.

3. $h_3(x_1||x_2||x_3||x_4) = h(x_1||x_2) \oplus h(x_3||x_4)$

Η h_3 δεν είναι collision resistant.

Πράγματι, θεωρώ string
μήκους 4n, τυχαίο $X = x_1||x_2||x_3||x_4$, με $x_1 \neq x_3$
μήκους 4n, $Y = x_3||x_4||x_1||x_2$

Είναι $h_3(X) = h(x_1||x_2) \oplus h(x_3||x_4)$

$h_3(Y) = h(x_3||x_4) \oplus h(x_1||x_2) = h_3(X)$

Και επειδή, από κατασκευή, είναι $x_1 \neq x_3$, σίγουρα $X \neq Y$.
Επομένως, βρήκαμε εύκολα σύγκρουση για την h_3.

4. $h_4(x_1||x_2||x_3||x_4) = h(h(h(x_1||x_2)||x_3)||x_4)$

Η h_4 είναι collision resistant.

Η λογική που ακολουθούμε για να το αποδείξουμε είναι αντίστοιχη αυτής του υποερωτήματος (2).

Block άτοπο: Έστω η h_2 δεν είναι collision resistant

- Μπορούμε να βρούμε εύκολα σύγκρουση για την h_4 , δηλαδή μπορούμε να βρούμε X, Y μήκους 4n με $X \neq Y$ και $h_4(X) = h_4(Y)$.
- Θεωρώ $X = x_1||x_2||x_3||x_4$, $h_4(X) = h(h(h(x_1||x_2)||x_3)||x_4)$
- Θεωρώ $Y = y_1||y_2||y_3||y_4$, $h_4(Y) = h(h(h(y_1||y_2)||y_3)||y_4)$
- $h(X) = h(Y) \Rightarrow h(h(h(x_1||x_2)||x_3)||x_4) = h(h(h(y_1||y_2)||y_3)||y_4)$
- Subblock άτοπο: Έστω $h(h(x_1||x_2)||x_3)||x_4 \neq h(h(y_1||y_2)||y_3)||y_4$
  - Θεωρώ $m_1 = h(h(x_1||x_2)||x_3)||x_4$, $m_2 = h(h(y_1||y_2)||y_3)||y_4$
  - Βρήκαμε m_1 , m_2 με $m_1 \neq m_2$ και $h(m_1) = h(m_2)$, δηλαδή βρήκαμε σύγκρουση για την h.

  Άτοπο, καθώς η h είναι collision resistant.

  Επομένως, $h(h(x_1||x_2)||x_3)||x_4 = h(h(y_1||y_2)||y_3)||y_4$
- Προκύπτει, $h(h(x_1||x_2)||x_3) = h(h(y_1||y_2)||y_3) \wedge x_4 = y_4$
- Όμοια με το προηγούμενο subblock άτοπο, δείχνουμε πως $h(x_1||x_2) = h(y_1||y_2) \wedge x_3 = y_3$
- Όμοια με το προηγούμενο subblock άτοπο, δείχνουμε πως $x_1||x_2 = y_1||y_2 \Rightarrow x_1 = y_1 \wedge x_2 = y_2$
- Άρα $x_1 = y_1 \wedge x_2 = y_2 \wedge x_3 = y_3 \wedge x_4 = y_4 \Rightarrow X = Y$

Άτοπο, καθώς, από κατασκευή $X \neq Y$.

Επομένως, η h_4 είναι collision resistant.

Άσκηση 5

Ο ορισμός της ψευδοτυχαίας συνάρτησης είναι ο εξής:
Έστω συνάρτηση F : {0, 1} × {0, 1}* → {0, 1}* αποδοτικά υπολογίσιμη, με κλειδί. Η F είναι ψευδοτυχαία συνάρτηση αν για κάθε πιθανοτικό πολυωνυμικού χρόνου διαχωριστή D υπάρχει αμελητέα συνάρτηση negl ώστε:*
$|Pr_k [D^{Fk()} (1^n) = 1] − Pr_f[D^{f()} (1^n) = 1]| \leq negl(n)$
όπου η πρώτη πιθανότητα είναι πάνω στην τυχαία επιλογή του κλειδιού $k \in \{0, 1\}^n$ και την τυχαιότητα του D, ενώ η δεύτερη ως προς την τυχαία επιλογή της $f \in Func_n$ και την τυχαιότητα του D.

Θεωρούμε πως ο διαχωριστής μπορεί να κάνει πολυωνιμικό πλήθος ερωτήσεων σε oracle, το οποίο αντιστοιχεί είτε στη συνάρτηση F_k ή σε ομοιόμορφα τυχαία επιλεγμένη $f \in Func_n$.

Στην άσκηση, δίίνεται $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1)^n$ ψευδοτυχαία συνάρτηση.

1. **$F_1(k, x) = F(k, x) \oplus x$**

   Η F_1 είναι ψευδοτυχαία συνάρτηση.

   Block άτοπο: Έστω η F_1 μη ψευδοτυχαία.
   - Υπάρχει διαχωριστής D ο οποίος με πολυωνυμικό αριθμό κλήσεων στο δοθέν oracle, διακρίνει κάποια ιδιότητα I στα string της εξόδου του oracle.
   - Η ιδιότητα I εμφανίζεται με
     πιθανότητα **$P_1 = Pr_k [D^{F1,k()} (1^n) = 1]$** στην έξοδο του $F_{1,k}$ oracle και
     πιθανότητα **$P_2 = Pr_f[D^{f()} (1^n) = 1]$** στην έξοδο του random oracle.
   - Εφόσον ο D πράγματι ξεχωρίζει την F_1 ως μη ψευδοτυχαία,
     είναι **$|P_1 - P_2| > negl(n)$** , $\forall\ negl_n()$
   - Ορίζουμε διαχωριστή D' για την F ο οποίος λειτουργεί με τον ίδιο τρόπο με τον D, με τη μόνη διαφορά πως ως πρώτο βήμα, στην έξοδο y του δοσμένου oracle υπολογίζει την ποσότητα $y' = y \oplus x$, όπου x η είσοδος του oracle και y(x) η έξοδος, και στη συνέχεια επιτελεί τη λειτουργία του D για το string y' .
   - Εάν το y προέκυψε από τυχαίο oracle, δηλαδή το y είναι ομοιόμορφα τυχαίο, τότε και το $y' = y \oplus x$ είναι ομοιόμορφα τυχαίο. Συνεπώς, το y' (από τυχαίο oracle) εμφανίζει την ιδιότητα I που εξετάζει ο D με πιθανότητα $P_2 = Pr_f[D^{f()} (1^n) = 1]$ (όπως και πριν), δηλαδή το y εμφανίζει την ιδιότητα I' που εξετάζει ο D' με πιθανότητα
     $Pr_f[D'^{f()} (1^n) = 1] = Pr_f[D^{f()} (1^n) = 1] = P_2$.
   - Τότε, ο διαχωριστής D' μπορεί να διακρίνει την F(k, x). Πράγματι, εάν πρέπει να διακρίνουμε ανάμεσα στην F(k,x) και την τυχαία συνάρτηση, τα string εξόδου της F(k,x) θα εμφανίζουν την ιδιότητα I' με πιθανότητα
     $Pr_k [D'^{Fk()} (1^n) = 1] = Pr_k [D^{F1,k()} (1^n) = 1] = P_1$,

καθώς $F_1(k, x) = F(k, x) \oplus x$, ενώ τα string εξόδου της τυχαίας θα εμφανίζουν την ιδιότητα I' με πιθανότητα P_2 , όπως προείπαμε, και ισχύει $|P_1 - P_2| > negl(n)$, $\forall\ negl_n()$

- Άρα, η F(k,x) δεν είναι ψευδοτυχαία.

Άτοπο, καθώς η F είναι ψευδοτυχαία.
Επομένως, η F_1 είναι ψευδοτυχαία.

2. $F_2(k, x) = F(F(k, 0^n), x)$

Η F_2 είναι ψευδοτυχαία συνάρτηση.

Πράγματι, δίνεται πως η F(k,x) είναι ψευδοτυχαία συνάρτηση.
Διαισθητικά, αυτό σημαίνει πως οι έξοδοι της F(k,x) για ομοιόμορφα τυχαίο επιλεγμένο k μοιάζουν να ακολουθούν ομοιόμορφη κατανομή από την οπτική ενός οποιουδήποτε διαχωριστή πολυωνυμικού χρόνου.
Η κατανομή αυτή φυσικά δεν ισχύει στην πραγματικότητα, αλλά είναι ικανή να "ξεγελάσει" οποιοδήποτε διαχωριστεί πολυωνυμικού χρόνου.
Συνεπώς, από αυτήν την οπτική, η επιλογή $F(k, 0^n)$ για ομοιόμορφα τυχαία επιλεγμένο k ισοδυναμεί με επιλογή "ομοιόμορφα τυχαίου" $k' = F(k, 0^n)$.

Τα παραπάνω φυσικά είναι απλά διαίσθηση και δε συνιστούν απόδειξη.
Το αποδεικνύουμε παρακάτω:

Block άτοπο: Έστω η F_1 μη ψευδοτυχαία.

- Υπάρχει διαχωριστής D ο οποίος με πολυωνυμικό αριθμό κλήσεων στο δοθέν oracle, διακρίνει κάποια ιδιότητα I στα string της εξόδου του oracle.
- Η ιδιότητα I εμφανίζεται με
  πιθανότητα $P_1 = Pr_k [D^{F2,k()} (1^n) = 1]$ στην έξοδο του $F_{2,k}$ oracle και
  πιθανότητα $P_2 = Pr_f[D^{f()} (1^n) = 1]$ στην έξοδο του random oracle.
- Εφόσον ο D πράγματι ξεχωρίζει την F_2 ως μη ψευδοτυχαία,
  είναι $|P_1 - P_2| > negl(n)$, $\forall\ negl_n()$
- Ορίζουμε διαχωριστή D' για την F ο οποίος λειτουργεί με τον εξής τρόπο. Καλεί το δοθέν oracle με είσοδο $x = 0^n$ και λαμβάνει έξοδο $y(0^n)$.
  Εάν το y προέκυψε από τυχαίο oracle, τότε $y(0^n) = f(0^n)$ για ομοιόμορφα τυχαία f.
  Εάν το y προέκυψε από το oracle F_k , τότε $y(0^n) = F(k, 0^n)$.
  Στη συνέχεια, ο διαχωριστής D', επιλέγει πολυωνυμικό πλήθος από x και εξετάζει τα string $y'(x) = F(y(0^n), x)$ για την ιδιότητα I (την οποία εξέταζε ο D).
- **Εάν $y(0^n) = F(k, 0^n)$**, τότε **$y'(x) = F(F(k,0^n), x) = F_{2,k}(x)$**,
  οπότε θα εμφανίζουν την ιδιότητα I με
  πιθανότητα **$Pr_k [D'^{Fk()} (1^n) = 1] = Pr_k [D^{F2,k()} (1^n) = 1] = P_1$**
- **Εάν $y(0^n) = f(0^n)$**, όπου f τυχαία
  τότε τα **$y'(x) = F(f(0^n), x)$**

και επειδή η f είναι τυχαία, το $f(0^n)$ είναι ένα ομοιόμορφα τυχαίο string μήκους n.
Συνεπώς, το y'(x) ισοδυναμεί με F(k', x), όπου k' ομοιόμορφα τυχαίο.
Οπότε τα **y'(x) = F(f(0^n), x)** εμφανίζουν την ιδιότητα I με
πιθανότητα **$Pr_f[D'^{f()}(1^n) = 1] = Pr_k[D^F(1^n) = 1] = P_2$**

- Είναι F(k, x) ψευδοτυχαία, άρα
  $|Pr_k[D'^{F_k()}(1^n) = 1] - Pr_f[D'^{f()}(1^n) = 1]| \le negl(n) \Rightarrow$
  $|P_1 - P_2| \le negl(n) \Rightarrow$
  $|Pr_k[D^{F2,k()}(1^n) = 1] - Pr_k[D^F(1^n) = 1]| \le negl(n) \Rightarrow$
  $|Pr_k[D^{F2,k()}(1^n)=1] - Pr_f[D^{f()}(1^n)=1] + Pr_f[D^{f()}(1^n)=1] - Pr_k[D^F(1^n)=1]| \le negl(n)$
  $||Pr_k[D^{F2,k()}(1^n)=1] - Pr_f[D^{f()}(1^n)=1]| - |Pr_f[D^{f()}(1^n)=1] - Pr_k[D^F(1^n)=1]|| \le negl(n)$
- Όμως, $|Pr_f[D^{f()}(1^n) = 1] - Pr_k[D^F(1^n) = 1]| \le negl(n)$, καθώς η F είναι ψευδοτυχαία
  και $|Pr_k[D^{F2,k()}(1^n) = 1] - Pr_f[D^{f()}(1^n) = 1]| > negl(n)$, καθώς υποθέσαμε πως ο D είναι ακριβώς ο διαχωριστής ο οποίος αναγνωρίζει την F_2.
  Επομένως, καθώς (non_negl + negl) ~ non_negl, προκύπτει πως
  $||Pr_k[D^{F2,k()}(1^n)=1] - Pr_f[D^{f()}(1^n)=1]| - |Pr_f[D^{f()}(1^n)=1] - Pr_k[D^F(1^n)=1]|| > negl(n)$

Άτοπο, καθώς προκύπτουν δύο αντιφατικές ανισότητες.
Επομένως, η F_1 είναι ψευδοτυχαία.

3. **$F_3(k, x) = F(F(k, 0^n), x)||F(k, x)$**

   Η F_3 δεν είναι ψευδοτυχαία.

   Θεωρώ το διαχωριστή πολυωνυμικού χρόνου D, ο οποίος ακολουθεί τα εξής βήματα:
   - Θέτουμε στο δοθέν oracle είσοδο 0^n, οπότε λαμβάνουμε έξοδο $y(0^n)$, μήκους 2n.
   - Θεωρώ **$y(0^n) = a(0^n)||b(0^n)$**, a, b μήκους n
   - Θεωρώ την πολυωνυμικά υπολογίσιμη ιδιότητα I για τις εξόδους του oracle:
     $a(x) = F(b(0^n), x)$, τα a(x), $b(0^n)$ εύκολα υπολογίσιμα
     και η F αποδοτικά υπολογίσιμη
   - Λοιπόν, εάν η εκάστοτε έξοδος **y(x) προέρχεται από το oracle της $F_{3,k}$** ,
     είναι **$y(x) = F(F(k, 0^n), x)||F(k, x) = a(x)||b(x)$**
     και **$b(0^n) = F(k, 0^n)$**
     οπότε $y(x) = F(b(0^n), x)||F(k,x) \Rightarrow a(x) = F(b(0^n), x)$
     δηλαδή η ιδιότητα I ισχύει με πιθανότητα $P_1 = 1$.
   - Εάν η έξοδος **y(x) προέρχεται από το τυχαίο oracle**, τότε
     με **y(x) = a(x)||b(x)**, τα a(x), b(x) είναι τυχαία
     οπότε τα a(x), $F(b(0^n), x)$ ταυτίζονται μόνο συμπτωματικά, δηλαδή η ιδιότητα I ικανοποιείται με πιθανότητα $P_2 = 2^{-n}$, καθώς πρόκειται για δυαδικά string μήκους n.
   - Είναι **$|P_1 - P_2| = 1 - 2^{-n}$**
   - Επομένως, η F_3 δεν είναι ψευδοτυχαία, καθώς το $1-2^{-n}$ δεν είναι negligible.

Άσκηση 6

1. $H_3(m) = H_1(m)||H_2(m)$

Η H_3 είναι collision resistant.

Block άτοπο: Έστω η H_3 δεν είναι collision resistant.

- Μπορούμε να βρούμε εύκολα σύγκρουση για την H_3 , δηλαδή μπορούμε να βρούμε x, y με $x \neq y$ και $H_3(x) = H_3(y)$.
- $H_3(x) = H_3(y) \Rightarrow H_1(x)||H_2(x) = H_1(y)||H_2(y)$
  $\Rightarrow H_1(x) = H_1(y) \wedge H_2(x) = H_2(y)$
  δηλαδή βρήκαμε σύγκρουση και για την H_1 και για την H_2

Άτοπο, καθώς τουλάχιστον μία εκ των H_1, H_2 είναι collision resistant.
Επομένως, η H_3 είναι collision resistant.

2. $H_4(m) = H_1(H_2(m))$

Δεν μπορούμε να αποφανθούμε γενικά για το αν η H_4 είναι collision resistant.
Διακρίνουμε 3 περιπτώσεις.

a. **Οι H_1, H_2 είναι και οι 2 collision resistant.**

   Σε αυτήν την περίπτωση, η H_4 είναι collision resistant.

   Block άτοπο: Έστω η H_4 δεν είναι collision resistant.
   - Μπορούμε να βρούμε εύκολα σύγκρουση για την H_4 , δηλαδή μπορούμε να βρούμε x, y με $x \neq y$ και $H_4(x) = H_4(y)$.
   - $H_3(x) = H_3(y) \Rightarrow H_1(H_2(x)) = H_1(H_2(y))$
   - Subblock άτοπο: Έστω $H_2(x) \neq H_2(y)$
     - Θεωρώ $m_1 = H_2(x)$, $m_2 = H_2(y)$
     - Βρήκαμε m_1 , m_2 με $m_1 \neq m_2$ και $H_2(m_1) = H_2(m_2)$,
       δηλαδή βρήκαμε σύγκρουση για την H_2.

     Άτοπο, καθώς η H_2 είναι collision resistant.
     Επομένως, $H_2(x) = H_2(y)$
   - Όμοια με το προηγούμενο subblock άτοπο, δείχνουμε πως x = y

   Άτοπο, καθώς, από κατασκευή, $x \neq y$.
   Επομένως, η H_4 είναι collision resistant.

b. Η H_1 είναι collision resistant, ενώ η H_2 δεν είναι.

Σε αυτήν την περίπτωση, η H_4 δεν είναι collision resistant.

Καθώς η H_2 δεν είναι collision resistant,
μπορώ εύκολα να βρω x, y με $x \neq y$ και $H_2(x) = H_2(y)$

Τότε, θεωρώ σταθερό αριθμό $c = H_2(x) = H_2(y)$.

Είναι $H_4(x) = H_1(H_2(x)) = H_1(c)$

$H_4(y) = H_1(H_2(y)) = H_1(c) = H_4(x)$

Και επειδή, από κατασκευή, $x \neq y$, βρήκαμε εύκολα σύγκρουση για την H_4 .

c. Η H_1 δεν είναι collision resistant, ενώ η H_2 είναι.

Σε αυτήν την περίπτωση, δεν μπορούμε να αποφανθούμε με σιγουριά για την H_4 .

Η H_4 δε θα είναι collision resistant αν μπορούμε να βρούμε εύκολα m_1, m_2 με $m_1 \neq m_2$ και τα $c_1 = H_2(m_1)$, $c_2 = H_2(m_2)$ να προκαλούν σύγκρουση στην H_1, δηλαδή $H_1(c_1) = H_1(c_2)$ (με $c_1 \neq c_2$) .

Για του λόγου το αληθές, θα δείξουμε δύο παραδείγματα που εμπίπτουν σε αυτήν την περίπτωση, όπου στο (i) η H_4 είναι collision resistant, ενώ στο (ii) δεν είναι.

(i). Έστω η H_1 αντιστοιχίζει κάθε άρτια είσοδο σε μια σταθερή έξοδο c, ενώ δεν εμφανίζει εύκολα συγκρούσεις για εισόδους m_1, m_2 όπου τουλάχιστον μία εξ αυτών είναι περιττή. Προφανώς, εμφανίζει συγκρούσεις για κάθε m_1, m_2 άρτια.
Αυτή είναι μια έγκυρη hash function (όχι collision resistant).

Επίσης, έστω η collision resistant H_2 αντιστοιχίζει όλες τις εισόδους σε περιττές εξόδους. Και αυτή είναι μια έγκυρη hash function.

Σε αυτό το παράδειγμα, η προκύπτουσα $H_4(m) = H_1(H_2(m))$ είναι collision resistant.

(ii). Έστω η H_1 αντιστοιχίζει κάθε είσοδο τετριμμένα σε μία σταθερή έξοδο c. Τότε, η H_1 εμφανίζει σύγκρουση για οποιεσδήποτε εισόδους.

Σε αυτό το παράδειγμα, (ανεξάρτητα από την επιλογή της H_2) η $H_4(m) = H_1(H_2(m))$ εμφανίζει σύγκρουση για οποιεσδήποτε εισόδους, οπότε, προφανώς, δεν είναι collision resistant.