- Konstantinos Filippou
- ics23044

Exercise 1

Η αριθμητική modulo αποτελεί τη βάση για πολλά κρυπτοσυστήματα. Υπολογίστε τα παρακάτω αποτελέσματα "στο χέρι" (χωρίς τη χρήση υπολογιστή):

- **7 * 14 mod 11**
- **7 * 14 mod 13**
- **4 * 3 mod 12**
- **-4 * 23 mod 21**
- **2^10 mod 2**

Solution

- 7 * 14 = 98 ----> 98 / 11 = 8 (πηλίκο), 98-8x11 = 98-88 = 10, Επομένως 7 * 14 mod 11 = 10
- 7 * 14 = 98 ----> 98 / 13 = 7 (πηλίκο), 98-7x11 = 98-91 = 7, Επομένως 7 * 14 mod 13 = 7
- 4 * 3 = 12 ----> 12 / 12 = 1 (πηλίκο), 12-1x12 = 12-12 = 0, Επομένως 4 * 3 mod 12 = 0
- -4 * 23 = -92 ----> -92 / 21 = -5 (πηλίκο), -92-(-5)x21 = -92+105 = 13, Επομένως -4 * 23 mod 21 = 13
- 2^10 = 1024 ----> 1024 / 2 = 512 (πηλίκο), 1024-512x2 = 1024-1024 = 0, Επομένως 2^10 mod 2 = 0

Exercise 2

Ποιος είναι ο πολλαπλασιαστικός αντίστροφος του 5 στους δακτυλίους Z5, Z25 και Z38? Παραμένει ο ίδιος ή αλλάζει με την αλλαγή του δακτυλίου? Έχει νόημα να αναφερόμαστε σε έναν αντίστροφο εάν δεν προσδιορίσουμε το modulo? Με ποιον αλγόριθμο γίνεται αποδοτικά ο υπολογισμός του αντιστρόφου σε έναν δακτύλιο?

Solution

Ο πολλαπλασιαστικός αντίστροφος του 5 στους δακτυλίους Z5, Z25 δεν υπάρχει καθώς ο μέγιστος κοινός διαιρέτης δεν είναι το 1 και είναι το 5. Στο Z38 ο μέγιστος κοινός διαιρέτης είναι το 1, επομένως υπάρχει πολλαπλασιαστικός αντίστροφος. Μέσω inverse συνάρτησης βγάζουμε τον αριθμό 23, (αφου s μέσω xgcd είναι το -15 και inverse of 5 mod 38= s+38 δηλαδή: -15+38= 23) και για να το επαληθεύσουμε αρκεί να δείξουμε ότι: $5\times23\equiv1 \mod 38$
5x23= 115 ---> 115/38 = 3 (πηλίκο), 115-3*38= 115-114= 1. Ο πολλαπλασιαστικός αντίστροφος αλλάζει με την αλλαγή του δακτυλίου αφού αλλάζει η πράξη mod που κάνουμε. Εάν δεν προσδιορίσουμε το Modulo δεν έχει νόημα να αναφερόμαστε σε έναν αντίστροφο καθώς είναι σχετικός ο ένας με τον άλλον. Ο αλγόριθμος που χρησιμοποιούμε για να υπολογίσουμε αποδοτικά τον αντίστροφο σε έναν δακτύλιο είναι ο Ευκλείδειος αλγόριθμος και συγκεκριμένα, η επέκταση του (Extended Euclidean Algorithm). Καθώς δεν υπολογίζει μόνο τον μέγιστο κοινό διαιρέτη αλλά εκφράζει το ΜΚΔ ως γραμμικό συνδυασμό των a και b, δηλαδή βρίσκουμε αριθμούς x και y, τέτοιοι ώστε:
$ΜΚΔ(a,b)=a\cdot x+b\cdot y$.
Αυτό είναι εξαιρετικά χρήσιμο όταν θέλουμε να βρούμε τον πολλαπλασιαστικό αντίστροφο του a modulo b, επειδή αν ΜΚΔ(a, b) = 1, τότε υπάρχει αντίστροφος και το x είναι ο αντίστροφος του a modulo b.

Exercise 3

1. Βρείτε όλους τους ακεραίους στο διάστημα **0 <= n < m** οι οποίοι είναι σχετικά πρώτοι του **m** , όπου **m = 9,17,25,33** . Ουσιαστικά θα πρέπει να αναζητήσετε το **φ(m)** για **m = 9,17,25,33** (Euler's phi function).

Solution

Για m = 9:

9 = 3^2 επομένως φ(9) = 9 x (1 - 1/3) = 9 x 2/3 = 6

Το 9 είχε πρώτο παράγοντα το 3

Τα πολλαπλάσια του 3 από [0,9) είναι το 3 και το 6

Άρα οι αριθμοί που είναι σχετικά πρώτοι με το 9 είναι οι: 1,2,4,5,7,8

Για m = 17:

Ο αριθμός 17 είναι πρώτος αριθμός επομένως φ(17) = 17-1= 16

Άρα οι αριθμοί που είναι σχετικά πρώτοι με το 17 είναι οι: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16

Για m = 25:

25 = 5^2 επομένως φ(25) = 25 x (1-1/5) = 25 x 4/5 = 20

Το 25 είχε πρώτο παράγοντα το 5

Τα πολλαπλάσια του 5 από [0,25) είναι το 5 , 10,15,20

Άρα οι αριθμοί που είναι σχετικά πρώτοι με το 25 είναι οι: 1,2,3,4,6,7,8,9,11,12,13,14,16,17,18,19,21,22,23,24

Για m = 33:

33 = 3*11 επομένως φ(33) = 33 x (1-1/3) x (1-1/11) = 33 x 2/3 x 10/11 = 33 x 20/33 = 20

Το 33 είχε πρώτο παράγοντα τους 3 και 11

Τα πολλαπλάσια του 3 από [0,33) είναι το 3 , 6,9,12,15,18,21,24,27,30

Τα πολλαπλάσια του 11 από [0,33) είναι το 11,22

Άρα οι αριθμοί που είναι σχετικά πρώτοι με το 33 είναι οι: 1,2,4,5,7,8,10,13,14,16,17,19,20,23,25,26,28,29,31,32

Exercise 4

Θεωρούμε τα σύνολα **Q = {q1,q2,...,q5}** και **P = {p1,p2,...,p5}** και τις πράξεις $\triangle$ και $\diamondsuit$ που ορίζονται από τους πίνακες:

$\triangle$	q_1	q_2	q_3	q_4	q_5
q_1	q_4	q_1	q_5	q_3	q_2
q_2	q_3	q_5	q_2	q_1	q_4
q_3	q_1	q_2	q_3	q_4	q_5
q_4	q_2	q_4	q_1	q_5	q_3
q_5	q_5	q_3	q_4	q_2	q_1

$\diamondsuit$	p_1	p_2	p_3	p_4	p_5
p_1	p_1	p_2	p_3	p_4	p_5
p_2	p_2	p_1	p_4	p_2	p_3
p_3	p_3	p_5	p_1	p_2	p_4
p_4	p_4	p_3	p_5	p_1	p_2
p_5	p_5	p_4	p_2	p_3	p_4

Εξηγήστε γιατί οι αλγεβρικές δομές $(Q, \triangle)$ και $(P, \diamondsuit)$ δεν είναι ομάδες.

Solution

Στην αλγεβρική δομή Q-Τρίγωνο , Είναι κλειστή η πράξη (όλα τα αποτελέσματα ειναι στο εύρος τιμών Q, ωστόσο παρατηρούμε ότι στην γραμμή q3 είναι q1,q2,q3,q4,q5 δηλαδή q3 $\triangle$ qi = qi, αντιθέτως στην στήλη δεν ισχύει το αντίθετο, π.χ q1 $\triangle$ q3 = q5 αντί για q1. Εφόσον δεν υπάρχει ουδέτερο στοιχείο - συνεπώς δεν ορίζονται αντίστροφα, τότε δεν είναι ομάδα.

Στην αλγεβρική δομή P-Ρόμβος, Είναι κλειστή η πράξη όπως στην Q , υπάρχει ουδέτερο στοιχείο p1 (δίνει πάντα pi) και ορίζονται αντίστροφα καθώς για καθε pi υπάρχει pj τετοιο ώστε pi $\diamondsuit$ pj = p1 = pj $\diamondsuit$ pi. Ωστόσο, δεν υπάρχει προσεταιριστικότητα καθώς αν πάρουμε (p2 $\diamondsuit$ p2) $\diamondsuit$ p3 = p3 ενώ p2 $\diamondsuit$ (p2 $\diamondsuit$ p3) = p2

Επομένως καμία απο τις δύο δεν είναι ομάδες.

Exercise 5

Θεωρήστε τον ομοπαραλληλικό κρυπταλγόριθμο με κλειδί **k=(k1,k2)** του οποίου οι συναρτήσεις κρυπτογράφησης και αποκρυπτογράφησης ορίζονται από τις εξισώσεις:

ek(m) = k1 * m + k2 (mod p)

dk(c) =k1^-1 * (c - k2) (mod p)

όπου p θετικός ακέραιος και **k1^-1** ο αντίστροφος του **k1 modulo p.**

- Έστω **p = 541** και ότι το κλειδί είναι **k = (34, 71)** . Κρυπτογραφήστε το μήνυμα **m = 204** . Αποκρυπτογραφήστε το κρυπτοκείμενο **c = 431** .
- Υποθέτοντας ότι το **p** είναι δημόσια γνωστό, εξηγήστε γιατί ο ομοπαραλληλικός κρυπταλγόριθμος είναι ευάλωτος σε μια **επίθεση επιλεγμένου απλού κειμένου**. Πόσα ζεύγη απλού κειμένου – κρυπτοκειμένου χρειάζονται ενδεχομένως προκειμένου να ανακτήσει ο αντίπαλος το κλειδί;
- Η Alice και ο Bob αποφασίζουν να χρησιμοποιήσουν τον πρώτο **p = 601** για τον ομοπαραλληλικό τους κρυπταλγόριθμο. Η τιμή του **p** είναι δημόσια γνωστή και η Eve υποκλέπτει τα κρυπτοκείμενα **c1 = 324** και **c2 = 381** , και καταφέρνει να ανακαλύψει ότι τα αντίστοιχα απλά κείμενα είναι **m1 = 387** και **m2 = 491**. Προσδιορίστε το κλειδί και στη συνέχεια χρησιμοποιήστε το για να κρυπτογραφήσετε το μήνυμα **m3 = 173** .

Solution

- ek(204) = 34 * 204 + 71 (mod 541) = 6936 + 71 (mod 541) = 7007 mod 541 ---> 7007 / 541 = 12 (πηλίκο), 7007 - 12 * 541 =7007 - 6492 = 515

ek(204) = 515

dk(431) = k1^-1 * (431 - 71) (mod 541) :

k1^-1 = k1 modulus p = 34 mod 541 = 541 = 34x15 + 31 // 34 = 31x1 +3 // 31= 3x10 + 1 // 3 = 1x3 + 0 (το υπόλοιπο έγινε 1 άρα αντιστρέφεται)

Αντιστρέφουμε όρους: 1 = 31 -3x10, 3 = 34-31 x 1 Άρα: 1 = 31 - (34-31)x10 = 31x11 - 34x10 και απο το πρώτο 31 = 541 - 34x15 αντικαθιστούμε:

1 = (541 - 34x15) x 11 - 34x10 = 541 x 11 - 34x(165+10) = 541x11 - 34x175. Άρα: -34x175 ≡1(mod541) οπότε $k1^{-1} \equiv 541 - 175 = 366$

(Αυτό γίνεται και με ένα απλό xgcd μέσω sagemath)

Επομένως η πράξη είναι 366x(360)(mod 541) = 131.760 mod 541 ---> 131.760/541 = 243(πηλίκο), 131.760 - 243 * 541 = 131.760 - 131.463 = 297

dk(431) = 297

- Ο ομοπαραλληλικός κρυπταλγόριθμος είναι ευάλωτος καθώς αρκεί ο αντίπαλος να επιλέξει αυθαίρετα γράμματα κειμένου και να παράγει τα αντίστοιχα κρυπτοκείμενα. Αρκεί ακριβώς δύο ζεύγη απλού κειμένου - κρυπτοκειμένου προκειμένου να ανακτήσει ο αντίπαλος το κλειδί, αρκεί η λύση της αφαίρεσης m1-m2 να είναι μοναδική.
- Αν πάρουμε το ek(m) έχουμε ότι ek(387) = 387k1 + k2 (mod 601) = 324 και ek(491) = 491k1 + k2 (mod 601) = 381

  Επομένως έχουμε ένα σύστημα και αν αφαιρέσουμε κατά μέλη προκύπτει ότι 104k1 = 57 modulus 601 βρίσκουμε τον αντίστροφο του 104 :

  601 = 104x5 + 81 , 104 = 81x1 + 23, 81 = 23x3 + 12, 23 = 12x1 + 11, 12 = 11x1 + 1, 11 = 1x11 + 0, (το υπόλοιπο έγινε 1 άρα αντιστρέφεται)

  Αντιστρέφουμε όρους:

  1 = 12-11x1, 11=23-12x1 Άρα 1=12-(23-12)= 12-23x1+12x1 = 2x12 -23, όμως απο πάνω έχουμε ότι 12=81-23x3 , άρα:

  1=2x(81-23x3)-23x1 = 2x81 - 6x23 - 23x1 = 2x81 - 7x23, όμως 23=104-81x1 , άρα: 1=2x81 - 7x(104-81) = 2x81 - 7x104 + 7x81 = 9x81 - 7x104 και

  81 = 601-104x5 και άρα προκύπτει το: 1=9x(601-104x5) - 7x104 = 9x601 - 45x104 - 7x104 = 9x601 - 52x104. Άρα 1=9x601 -52x104 και επειδή είμαστε στο modulus 601, 1=-52x104 , άρα x=-52 και επειδή είναι αρνητικός το inverse είναι: 601-52 = 549.

  (γίνεται και με inverse function του εργαστηρίου `inverse(104,601)`) στη συνέχεια έχουμε: 549 * 57 = 549

  31293 modulo 601 = 31293/601 = 52 (πηλίκο), 31293 - 52*601 = 31293 - 31252 = 41. Άρα k1 = 41 και για k2 από πρώτη σχέση: 387 * 41 + k2 = 324 (mod 601) ,

  15867 + k2 = 324 (mod 601), 15867 mod 601 ---> 15867/601 = 26 (πηλίκο), 15867 - 26*601 = 15867 -

15626 = 241. Άρα 241 + k2 = 324, επομένως k2 = 83. Άρα το κλειδί ειναι (41,83). Για m3 = 173 έχουμε: ek(173) = 173x41 + 83 (modulus 601) = 7093 + 83 (modulus 601) = 7176 mod 601--> 7176/601 =11

7176 - 11*7176 = 7176 - 6611 = 565, Επομένως c3 = 565

Exercise 6

- Εξεστάστε τη συμπεριφορά του ομοπαραλληλικού κρυπταλγορίθμου σε **επίθεση γνωστού απλού κειμένου** έχοντας κατανοήσει τη διαφορά των επιθέσεων CPA και KPA: - περιγράψτε τη συνθήκη που πρέπει να ισχύει για το μήνυμα απλού κειμένου προκειμένου η επίθεση να είναι επιτυχής.
- η επιλογή των γραμμάτων **E** και **L** των οποίων το αντίστοιχο κρυπτοκείμενο είναι τα γράμματα **M** και **V** αποκαλύπτει το κλειδί? Αν ναι, ποιο είναι αυτό?
- η γνώση ότι τα γράμματα **M** και **Y** παράγουν ως κρυπτοκείμενο τα γράμματα **R** και **V** αποκαλύπτει το κλειδί? Τεκμηριώστε κατάλληλα.
- έχοντας κατανοήσει τη διαφορά των επιθέσεων CPA και KPA, βελτιώστε τον κώδικα της συνάρτησης **affine_analysis** ώστε να δίνει την κατάλληλη έξοδο κάθε φορά.

Solution

- Η συνθήκη που πρέπει να ισχύει είναι επαναχρησιμοποίηση του κλειδιού K σε περισσότερα από ένα ciphertext και η γνώση τουλάχιστον ενός αντίστοιχου plaintext διότι αν: C1=P1⊕K και C2 = P2⊕ K τότε από το γνωστό ζεύγος προκύπτει K = C1 ⊕ P1. Έχοντας το K και βλέποντας το C2 ο επιδρομέας αποκτά αμέσως το P2=C2⊕K.
- m1 = 4 (γράμμα E) , m2 = 11 (γράμμα L) , c1 = 12 (γράμμα M) c2 = 21 (γράμμα V) και έχουμε τις συναρτήσεις:

  c1 = k1m1 + k2 (mod 26)

  c2 = k1m2 + k2 (mod 26) οι οποίες γίνονται:

  12 = 4k1 + k2 (1)

  21 = 11k1 + k2 (2)

  Αν αφαιρέσουμε κατά μέλη γίνεται: 7k1 = 9 (mod 26) βρίσκουμε το αντίστροφο του 7 ,

  26 = 7x3 + 5, 7=5x1 + 2, 5=2x2 + 1, 2=1x2 + 0, άρα αντιστρέφεται άρα υπάρχει κλειδί:

  1 = 5-2x2 , 2=7-5x1, άρα 1=5-2x(7-5x1) = -2x7+5x3 , και έχουμε ότι 5=26-7x3 ,άρα 1=-2x7+3x(26-7x3) = -2x7+3x26-7x9

  1 = -11x7+3x26 , 1=-11x7 , 26-11 = 15, άρα ο αντίστροφος του 7 στο mod 26 είναι το 15. Λύνουμε για k1:

  7k1 = 9 (mod 26), k1 ≡ 9·7^-1 ≡9·15=135 (mod 26) , 135 mod 26 = 135/26 = 5 (πηλίκο), 135-5x26 = 135-130 = 5 , k1 = 5,

  και κ2 απο σχέση (1) => 12=4x5 + k2 => k2 = 12-20 = -8 , 26-8 = 18, άρα κ2= 18
- m1 = 12 (γράμμα M) , m2 = 24 (γράμμα Y) , c1 = 17 (γράμμα R) c2 = 21 (γράμμα V) και έχουμε τις συναρτήσεις:

  c1 = k1m1 + k2 (mod 26)

  c2 = k1m2 + k2 (mod 26) οι οποίες γίνονται:

  17 = 12k1 + k2 (1)

  21 = 24k1 + k2 (2)

Αν αφαιρέσουμε κατά μέλη γίνεται: 8k1 = 4 (mod 26) βρίσκουμε το αντίστροφο του 8 ,

26 = 8x3 + 2, 8=2x4 + 0 , δεν έχει υπόλοιπο 1 άρα δεν αντιστρέφεται άρα δεν υπάρχει κλειδί.

- Η μόνη αλλάγη που χρειάζεται είναι ένας έλεγχος αν το key[0] αντιστρέφεται ή όχι μέσω της συνάρτησης gcd. Παρακάτω έχουμε ένα test case με σωστά δεδομένα και ένα με λάθος (τα παραπάνω δεδομένα τις άσκησης)

(1)

[39]:
```
# επίθεση επιλεγμένου απλού κειμένου - CPA
n=26
Z26=IntegerModRing(n)

plaintext='EL'
ciphertext='MV'

def affine_analysis(m,c):
    mList=str2lst(m)
    cList=str2lst(c)

    A = matrix(Z26, 2, 2, [mList[0], 1, mList[1], 1])
    b = vector(Z26, [cList[0], cList[1]])
    key=A.solve_right(b)
    return key

key=affine_analysis(plaintext,ciphertext)
if (gcd(key[0],26) == 1 ):
    print('The key is:', key)
    print("Verification: the ciphertext message: " + ciphertext + " is decrypted in --> " + affine_dec(ciphertext,int(key[0]),int(key[1])))
else:
    print("No keys available.")
```

```
The key is: (5, 18)
Verification: the ciphertext message: MV is decrypted in --> EL
```

(2)

[40]:
```
# επίθεση επιλεγμένου απλού κειμένου - CPA
n=26
Z26=IntegerModRing(n)

plaintext='EL'
ciphertext='RV'

def affine_analysis(m,c):
    mList=str2lst(m)
    cList=str2lst(c)

    A = matrix(Z26, 2, 2, [mList[0], 1, mList[1], 1])
    b = vector(Z26, [cList[0], cList[1]])
    key=A.solve_right(b)
    return key

key=affine_analysis(plaintext,ciphertext)
if (gcd(key[0],26) == 1 ):
    print('The key is:', key)
    print("Verification: the ciphertext message: " + ciphertext + " is decrypted in --> " + affine_dec(ciphertext,int(key[0]),int(key[1])))
else:
    print("No keys available.")
```

```
No keys available.
```