- Konstantinos Filippou
- ics23044

Exercise 1

Δημιουργήστε τον 8×8 πολλαπλασιαστικό πίνακα για το σώμα επέκτασης $GF(2^3)$ έχοντας ως ανάγωγο πολυώνυμο το $P(x) = x^3 + x^2 + 1$.

Solution

```
K.<a> = GF(2)[]
F.<x> = GF(2^3, modulus=x^3 + x^2 + 1)
elements = list(F)
mult_table = [[a*b for b in elements] for a in elements]

for row in mult_table:
    print(row)
```

```
[0, 0, 0, 0, 0, 0, 0, 0]
[0, x^2, x^2 + 1, x^2 + x + 1, x + 1, x^2 + x, 1, x]
[0, x^2 + 1, x^2 + x + 1, x + 1, x^2 + x, 1, x, x^2]
[0, x^2 + x + 1, x + 1, x^2 + x, 1, x, x^2, x^2 + 1]
[0, x + 1, x^2 + x, 1, x, x^2, x^2 + 1, x^2 + x + 1]
[0, x^2 + x, 1, x, x^2, x^2 + 1, x^2 + x + 1, x + 1]
[0, 1, x, x^2, x^2 + 1, x^2 + x + 1, x + 1, x^2 + x]
[0, x, x^2, x^2 + 1, x^2 + x + 1, x + 1, x^2 + x, 1]
```

Exercise 2

Πρόσθεση στο σώμα επέκτασης $GF(2^4)$: υπολογίστε το $(A(x) + B(x))modP(x)$ στο $GF(2^4)$ χρησιμοποιώντας το ανάγωγο πολυώνυμο $P(x) = x^4 + x^3 + 1$, όταν:

- $A(x) = x^2 + 1, B(x) = x^3 + x^2 + 1$
- $A(x) = x^2 + 1, B(x) = x + 1$

Ποια είναι η επίδραση της επιλογής του πολυωνύμου $P(x)$ στον υπολογισμό του αθροίσματος δύο πολυωνύμων?

Solution

- A(x) = x^2 +1

  B(x) = x^3 + x^2 + 1

  C(x) = x^3
- A(x) = x^2 + 1

  B(x) = x + 1

  C(x) = x^2 + x

Η επιλογή του πολυωνύμου δεν επηρεάζει κάπου στον υπολογισμό του αθροίσματος καθώς όλα απλώς γίνονται modulo 2.

Exercise 3

Πολλαπλασιασμός στο $GF(2^4)$: υπολογίστε το $(A(x) \times B(x)) mod P(x)$ στο $GF(2^4)$ χρησιμοποιώντας το ανάγωγο πολυώνυμο $P(x) = x^4 + x^3 + 1$, όταν:

- $A(x) = x^2 + 1, B(x) = x^3 + x^2 + 1$
- $A(x) = x^2 + 1, B(x) = x + 1$

Ποια είναι η επίδραση της επιλογής του πολυωνύμου $P(x)$ στον υπολογισμό του γινομένου δύο πολυωνύμων?

Solution

- A(x) = x^2 +1

  B(x) = x^3 + x^2 + 1

  C(x) = (x^2+1) * (x^3+x^2+1) = x^5 + x^4 + x^2 + x^3 + x^2 + 1 = x^5 + x^4 + x^3 + 1

  Με αναγωγή στο modulo P(x) γίνεται:

  Ξέρουμε από το P(x) ότι x^4 = x^3 + 1 Επομένως γίνεται:

  x^4 + x + x^4 + x^3 + 1 = (x^4 + x^4) + x^3 + x + 1 = x^3 + x + 1

- A(x) = x^2 + 1

  B(x) = x + 1

  C(x) = (x^2 + 1) * (x+1) = x^3 + x^2 + x + 1δεν χρειάζεται αναγωγή αφού ο βαθμός πολυ/μου μικρότερος του 4.

  Η επίδραση είναι ότι αναλόγως με τον βαθμό του πολυωνύμου , επιλέγουμε αν θα κάνουμε αναγωγή ή όχι. Στην πρώτη περίπτωση στη μία χρειάστηκε καθώς ο βαθμός ήταν μεγαλύτερος του 4 και στην δεύτερη δεν χρειάστηκε.

Exercise 4

Να βρεθούν οι πολλαπλασιαστικοί αντίστροφοι στο $GF(2^8)$ των: 91, C3, DA, 6B, και 22 με τη βοήθεια του S-Box του AES. Στη συνέχεια να γίνει επαλήθευση στο Sage.

Solution

Για (91)hex: ψάχνω στον πίνακα την γραμμή 9 και την στήλη 1 και βρίσκω το (6A)hex = (01101010)binary = x^6 + x^5 + x^3 + x

Για (C3)hex: ψάχνω στον πίνακα την γραμμή C και την στήλη 3 και βρίσκω το (A3)hex = (10100011)binary = x^7 + x^5 + x + 1

Για (DA)hex: ψάχνω στον πίνακα την γραμμή D και την στήλη A και βρίσκω το (C4)hex = (11000100)binary = x^7 + x^6 + x^2

Για (6B)hex: ψάχνω στον πίνακα την γραμμή 6 και την στήλη B και βρίσκω το (DF)hex = (11011111)binary = x^7 + x^6 + x^4 + x^3 + x^2 + x + 1

Για (22)hex: ψάχνω στον πίνακα την γραμμή 2 και την στήλη 2 και βρίσκω το (5A)hex = (01011010)binary = x^6 + x^4 + x^3 + x

Για την επαλύθευση αναπτύσσουμε τον αντίστοιχο κώδικα:

```
K.<a> = GF(2)[]
F.<x> = GF(2^8, modulus=a^8 + a^4 + a^3 + a + 1)

print("Irreducible polynomial:", F.modulus())
print()

hex_vals = [0x91, 0xC3, 0xDA, 0x6B, 0x22]

def int_to_GF256(n):
    return sum(((n >> i) & 1) * x^i for i in range(8))

elements = [int_to_GF256(v) for v in hex_vals]

inverses = [e^-1 for e in elements]

for h, e, inv in zip(hex_vals, elements, inverses):
    print("Element:", hex(h))
    print("As polynomial:", e)
    print("Inverse:", inv)
    print("Check:", e * inv)
    print()
```

και τα αποτελέσματα:

```
Irreducible polynomial: x^8 + x^4 + x^3 + x + 1

Element: 0x91
As polynomial: x^7 + x^4 + 1
Inverse: x^6 + x^5 + x^3 + x
Check: 1

Element: 0xc3
As polynomial: x^7 + x^6 + x + 1
Inverse: x^7 + x^5 + x + 1
Check: 1

Element: 0xda
As polynomial: x^7 + x^6 + x^4 + x^3 + x
Inverse: x^7 + x^6 + x^2
Check: 1

Element: 0x6b
As polynomial: x^6 + x^5 + x^3 + x + 1
Inverse: x^7 + x^6 + x^4 + x^3 + x^2 + x + 1
Check: 1

Element: 0x22
As polynomial: x^5 + x
Inverse: x^6 + x^4 + x^3 + x
Check: 1
```

Exercise 5

Θεωρούμε το Προηγμένο Πρότυπο Κρυπτογράφησης (AES) με μήκος τμήματος 128-bit και κλειδί μήκους 128-bit. Ποια είναι η έξοδος του πρώτου γύρου του AES για είσοδο (state) αποτελούμενη από 128 άσσους και κλειδί γύρου να αποτελείται επίσης από 128 άσσους? Μπορείτε να αναπαραστήσετε τα δεδομένα και τα αποτελέσματα ως στοιχεία του δεκαεξαδικού συστήματος οργανωμένα σε πίνακα 4×4.

Solution

State (εισόδου): 128 bits = 16 bytes, όλα 0x01

Κλειδί γύρου (round key): 128 bits = 16 bytes, όλα 0x01

State =

01	**01**	**01**	**01**
01	01	01	01
01	01	01	01
01	01	01	01

Round Key =

01	**01**	**01**	**01**
01	01	01	01
01	01	01	01
01	01	01	01

AddRoundKey

Το AES ξεκινά με KeyAddition, δηλαδή XOR κάθε byte του state με το αντίστοιχο byte του κλειδιού γύρου:

Επομένως οι πίνακες γίνονται έτσι:

State =

00	**00**	**00**	**00**
00	00	00	00
00	00	00	00
00	00	00	00

Round Key =

00	**00**	**00**	**00**
00	00	00	00
00	00	00	00
00	00	00	00

Στη συνέχεια, ακολουθούμε το βήμα του ByteSub, και κάνουμε αντικατάσταση κάθε byte με τον αντίστοιχο πίνακα S-box και μετατρέπεται σε 0x63 και στο ShiftRows όλα τα bytes είναι ίδια οπότε δεν αλλάζει κάτι.

Στο MixColumn εκτελούμε πολλαπλασιασμό πινάκων στο GF(2^8). Κάθε στήλη του πίνακα που έχουμε πολλαπλασιάζεται με τον πίνακα του MixColumn στο AES.

2 3 1 1
1 2 3 1 πολλαπλασιάζεται με 63.
1 1 2 3
3 1 1 2

Επομένως γίνεται:

(2 * 0x63) XOR (3 * 0x63) XOR (1 * 0x63) XOR (1 * 0x63)

- (1 * 0x63) XOR (1 * 0x63) = 00

- (2 * 0x63) XOR (3 * 0x63) = (2 * 0x63) XOR (2 * 0x63) XOR (1 * 0x63) = 00 XOR (1 * 0x63)

  63 XOR 00 = 63

και εφόσον κάθε στοιχείο έχει την ίδια τιμή παραμένει ίδιο.

Τέλος προσθέτουμε το κλειδί του πρώτου γύρου που αποτελείται από άσσους.

Επομένως κάνουμε 63 XOR 01

0x63 = 01100011

0x01 = 00000001

= 01100010 = 0x62

Άρα το τελικό byte είναι 62.

Άρα ο πίνακας θα είναι ο εξής:

62	**62**	**62**	**62**
62	62	62	62
62	62	62	62
62	62	62	62