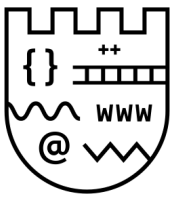


Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Τμήμα Πληροφορικής

Τεχνική αναφορά για NGE-06-03

Συμμετρική Κρυπτογραφία

Αλέξανδρος Κόρκος
alexkork@csd.auth.gr
3870

Θεσσαλονίκη, 10 Μαΐου 2023

Περιεχόμενα

1 Σύστημα μετατόπισης

1.1 Εκφώνηση

Το επόμενο κρυπτόγραμμα έχει ληφθεί:

οκηθμφδζθγοθχυκχσφθμφμχγ

Ο αλγόριθμος κρυπτογράφησης είναι ο εξής:

Κάθε γράμμα του αρχικού μας μηνύματος αντικαθίσταται από την αριθμητική του τιμή (α: 1, ..., ω: 24). Ας είναι x_0 μία ρίζα του τριωνύμου $g(x) = x^2 + 3x + 1$. Σε κάθε αριθμό του μηνύματός μου προσθέτω την τιμή του πολυωνύμου $f(x) = x^5 + 3x^4 + 3x^3 + 7x^2 + 5x + 4$, στο x_0. Αντικαθιστώ κάθε αριθμό με το αντίστοιχο γράμμα. Βρείτε το αρχικό μήνυμα.

1.2 Λύση

1.2.1 Μαθηματική επίλυση

Από την εκφώνηση, γνωρίζω ότι $g(x_0) = 0$.

Επίσης, θα απλοποιήσω την $f(x)$ μέσω της διαίρεση πολυωνυμικών συναρτήσεων έτσι ώστε να εκμεταλλευτώ την παραπάνω πληροφορία. Συνεπώς:

$$f(x) = g(x) \cdot h(x) \Rightarrow h(x) = \frac{f(x)}{g(x)} \Rightarrow h(x) = x^3 + 2x + 1 + \frac{3}{x^2 + 3x + 1} \quad (1)$$

Κάνοντας εφαρμογή της επιμεριστικής ιδιότητας στην (1), προκύπτει:

$$f(x) = (x^3+2x+1)\cdot g(x) + \frac{3}{\cancel{x^2 + 3x + 1}} \cdot \cancel{g(x)} \Rightarrow f(x) = (x^3+2x+1)\cdot g(x)+3 \quad (2)$$

Εξετάζω την συνάρτηση (2) για $x = x_0$:

$$f(x_0) = (x_0^3 + 2x_0 + 1) \cdot \cancel{g(x_0)}^{\,0} + 3 \Rightarrow f(x_0) = 3 \quad (3)$$

Άρα, όλα τα γράμματα της αλφάβητα έχουν μετατοπιστεί κατά 3 θέσεις.

1.2.2 Λεπτομέρειες για την υλοποίηση

Εκτέλεση αρχείου:

```
python shift.py
```

Μια απλή υλοποίηση του συστήματος μετατόπισης, δηλαδή της συνάρτησης:

$$D_n(x) = x - n \pmod{24}, n = 3 \quad (4)$$

1.2.3 Συμπέρασμα

Εκτελώντας το αρχείο για το μηνύματος εκφώνησης επιστρέφεται το εξής αποκρυπτογραφημένο μήνυμα:

```
Μηδεις αγεωμετρητος εισιτω
```

όπου ήταν φράση που βρισκόταν στην είσοδο της ακαδημίας του Πλάτωνα [1].

2 Το σύστημα του Vigenere

2.1 Εκφώνηση

Αποκρυπτογραφήστε το κείμενο [2], που κρυπτογραφήθηκε με τον αλγόριθμο του Vigenere.

Υποδ. Για την αποκρυπτογράφηση συστήνουμε να χρησιμοποιήσετε python. Για το μήκος του κλειδιού μπορείτε να χρησιμοποιήσετε έιτε test Kasiski ή την μέθοδο του Friedman.

2.2 Λύση

2.2.1 Λεπτομέρειες για την υλοποίηση

Βιβλιοθήκες

- functools
- re

```
python vegenere.py <όνομα_αρχείου>
```

Ως όνομα αρχείου δίνεται το αρχείο που θέλουμε να αποκρυπτογραφήσουμε.

Αρχικά, η βασικότερη συνάρτηση είναι η `kasiski`, που υλοποιεί την μέθοδο, σύμφωνα με [2]. Για να επιτευχθεί η frequency analysis, υλοποιείται η συνάρτηση `frequencyAnalysis` που βασίζεται στο άρθρο [1].

2.2.2 Συμπέρασμα

Για να βρεθεί το αποκρυπτογραφημένο μήνυμα, διακρίνεται σε 3 βήματα που περιγράφονται στην συνέχεια.

Εκτελώντας λοιπόν το τεστ Kasiski, βρίσκετε πως το μήκος του κλειδιού είναι 7 χαρακτήρες. Για να φτάσουμε σε αυτό τον αριθμό, δοκιμάσθηκε η

[1] Πηγή
[2] Μέθοδος Kasiski

ομαδοποίηση του κειμένου με διαφορά πλήθη χαρακτήρων. Για ομαδοποίηση ≥ 5 χαρακτήρων καταλήγουμε στο μήκος κλειδιού $= 7$. Για < 5 καταλήγουμε σε κλειδιά μήκους είτε 1 είτε 2 χαρακτήρων, που δεν αποκρυπτογραφούν το μήνυμα.

Στην συνέχεια, γίνετε ανάλυση συχνότητας γλώσσας (frequency analysis) για μήκος κλειδιού $= 7$. Η λέξη κλειδί στην οποία καταλήγουμε είναι SHANNON.

Τέλος, κάνοντας αποκρυπτογράφηση με λέξη κλειδί SHANNON προκύπτει το παρακάτω απόσπασμα μηνύματος (σχήμα 1) και συμπεραίνετε πως το κλειδί είναι ορθό μιας και το αποκρυπτογραφημένο μήνυμα απαρτίζεται από αγγλικές λέξεις.

```
A VERY SMALL PERCENTAGE OF THE POPULATION PRODUCES THE
GREATEST PROPORTION OF THE IMPORTANT IDEAS THIS IS A KIN
TO AN IDEA PRESENTED BY AN ENGLISH MATHEMATICIAN TURING
THAT THE HUMAN BRAIN IS SOMETHING LIKE A PIECE OF URANIUM
THE HUMAN BRAIN IF IT IS BELOW THE CRITICAL LAP AND YOU
                    SHOOT ONE ...
```

Σχήμα 1: Απόσπασμα του αποκρυπτογραφημένου μηνύματος

3 Κυκλική κύλιση

3.1 Εκφώνηση

Έστω ένα μήνυμα m, 16-bits. Θεωρούμε την κυκλική κύλιση προς τα αριστερά $\ll \alpha$ κατά α bits. Έστω ότι m κωδικοποιείται στο c σύμφωνα με τον τύπο,

$$c = m \oplus (m \ll 6) \oplus (m \ll 10) \tag{5}$$

Βρείτε τον τύπο αποκωδικοποίησης. Δηλαδή, γράψτε το m ως συνάρτηση του c. Υλοποιήστε κατάλληλο κώδικα για να δείξετε ότι ο τύπος που φτιάξατε είναι σωστός.

3.2 Λύση

3.2.1 Μαθηματική επίλυση

Γνωρίζω πως:

$$m = \begin{pmatrix} m_0 & m_1 & m_2 & m_3 & m_4 & \dots & m_{11} & m_{12} & m_{13} & m_{14} & m_{15} \end{pmatrix}$$

$$m \ll 6 = \begin{pmatrix} m_6 & m_7 & m_8 & m_9 & m_{10} & \dots & m_4 & m_3 & m_2 & m_1 & m_0 \end{pmatrix}$$

$$m \ll 10 = \begin{pmatrix} m_{10} & m_{11} & m_{12} & m_{13} & m_{14} & \dots & m_4 & m_3 & m_2 & m_1 & m_0 \end{pmatrix}$$

συνεπώς, έτσι μπορώ να ορίσω τον "πίνακα" c όπου κάθε σειρά περιλαμβάνει μια πράξη XOR μεταξύ αυτών.

$$c = \begin{pmatrix} m_0 & m_1 & m_2 & m_3 & m_4 & \dots & m_{11} & m_{12} & m_{13} & m_{14} & m_{15} \\ m_6 & m_7 & m_8 & m_9 & m_{10} & \dots & m_4 & m_3 & m_2 & m_1 & m_0 \\ m_{10} & m_{11} & m_{12} & m_{13} & m_{14} & \dots & m_4 & m_3 & m_2 & m_1 & m_0 \end{pmatrix} \quad (6)$$

Για αποκωδικοποιήσουμε το αρχικό μήνυμα, θα πρέπει να φθάσω $c \to m$ μέσω κάποιων πράξεων XOR και Left - Shift. Αρχικά, θα γίνει $c \oplus c \ll 10$ ε.ω. να απαλείψω την πρώτη και τρίτη "σειρά" του c καθώς θα είναι κοινές και στους δυο "πίνακες".

$$c \ll 10 = \begin{pmatrix} m_{10} & m_{11} & m_{12} & m_{13} & m_{14} & \dots & m_5 & m_6 & m_7 & m_8 & m_9 \\ m_0 & m_1 & m_2 & m_3 & m_4 & \dots & m_{11} & m_{12} & m_{13} & m_{14} & m_{15} \\ m_4 & m_5 & m_6 & m_7 & m_8 & \dots & m_{15} & m_0 & m_1 & m_2 & m_3 \end{pmatrix}$$

$$c_1 = c \oplus (c \ll 10) \quad (7)$$

$$(7) \Rightarrow c_1 = \begin{pmatrix} m_6 & m_7 & m_8 & m_9 & m_{10} & \dots & m_4 & m_3 & m_2 & m_1 & m_0 \\ m_4 & m_5 & m_6 & m_7 & m_8 & \dots & m_{15} & m_0 & m_1 & m_2 & m_3 \end{pmatrix}$$

Θα απαλείψω στην συνέχεια, την πρώτη σειρά του c_1 κάνοντάς XOR με τον c_1 μετατοπισμένο κατά 2.

$$c_1 \ll 2 = \begin{pmatrix} m_8 & m_9 & m_{10} & m_{11} & m_{12} & \dots & m_3 & m_4 & m_5 & m_6 & m_7 \\ m_6 & m_7 & m_8 & m_9 & m_{10} & \dots & m_4 & m_3 & m_2 & m_1 & m_0 \end{pmatrix}$$

$$c_2 = c_1 \oplus (c_1 \ll 2) \quad (8)$$

$$(8) \Rightarrow c_2 = \begin{pmatrix} m_8 & m_9 & m_{10} & m_{11} & m_{12} & \dots & m_3 & m_4 & m_5 & m_6 & m_7 \\ m_4 & m_5 & m_6 & m_7 & m_8 & \dots & m_{15} & m_0 & m_1 & m_2 & m_3 \end{pmatrix}$$

Έχοντας τώρα στον "πίνακα" c_2 2 σειρές, μπορώ μια μια μετατοπισμένη έκδοση του c να τον φέρω σε μοναδιαία σειρά.

$$c \ll 14 = \begin{pmatrix} m_{14} & m_{15} & m_0 & m_1 & m_2 & \dots & m_9 & m_{10} & m_{11} & m_{12} & m_{13} \\ m_8 & m_9 & m_{10} & m_{11} & m_{12} & \dots & m_3 & m_4 & m_5 & m_6 & m_7 \\ m_4 & m_5 & m_6 & m_7 & m_8 & \dots & m_{15} & m_0 & m_1 & m_2 & m_3 \end{pmatrix}$$

$$c_3 = c_2 \oplus (c \ll 14) \tag{9}$$

$$(9) \Rightarrow c_3 = \begin{pmatrix} m_{14} & m_{15} & m_0 & m_1 & m_2 & \dots & m_9 & m_{10} & m_{11} & m_{12} & m_{13} \end{pmatrix}$$

Τέλος, μπορώ να δω ότι εάν μετατοπίσω τον c_3 κατά 2, θα έχω το m.

$$m = c_3 \ll 2 \tag{10}$$

$$(10) \Rightarrow m = \begin{pmatrix} m_0 & m_1 & m_2 & m_3 & m_4 & \dots & m_{11} & m_{12} & m_{13} & m_{14} & m_{15} \end{pmatrix}$$

3.2.2 Λεπτομέρειες για την υλοποίηση

Εκτέλεση αρχείου:

```
python cyclic_shift.py
```

Έχουνε δημιουργηθεί τρεις συναρτήσεις. Μια που υλοποιεί την πράξη XOR, μια την πράξη Left - Shift και μια που παράγει τυχαία δυαδικά μηνύματα.

Αφού παραχθεί πρώτα ένα τυχαίο δυαδικό μήνυμα μήκους 16-bits, κρυπτογραφειται με την μέθοδο της εκφώνησης και στην συνέχεια αποκρυπτογραφείται με την μέθοδο που αναπτύχθηκε προηγουμένως.

3.2.3 Συμπέρασμα

Μπορούμε να αποφανθούμε, πως η μέθοδος αποκρυπτογράφησης αποτελεί ένα τρόπο για να βρούμε το αρχικό μήνυμα, καθώς ακολουθώντας διαφορετική συλλογιστική πορεία στις πράξεις μπορούμε και πάλι να βρούμε το αρχικό μήνυμα.

4 Τέλεια ασφάλεια

4.1 Εκφώνηση

Να αποδείξετε ότι, αν στο σύστημα μετατόπισης διαλέγουμε τυχαία τα κλειδιά από το σύνολο $\{0, 1, ..., 23\}$, τότε το σύστημα έχει τέλεια ασφάλεια.

4.2 Λύση

4.2.1 Μαθηματική επίλυση

Για να έχει ένα σύστημα τέλεια ασφάλεια (Perfect Secrecy) θα πρέπει κατά C. Shannon να ισχύει το εξής (με την βοήθεια των σημειώσεων της διάλεξης [2]):

Ορισμός. *Έστω $(\mathcal{M}, \mathcal{C}, \mathcal{K}, E, D)$ έχει τέλεια ασφάλεια, αν $\forall(m_1, m_2) \in \mathcal{M}$ και για $c \in \mathcal{C}$ ισχύει:*

$$Pr[k \leftarrow \mathcal{K} : E(m_1, k) = c] = Pr[k \leftarrow \mathcal{K} : E(m_2, k) = c], \forall k \xleftarrow{\$} \mathcal{K} \tag{11}$$

Θα εξετασθεί για αρχή, τα μηνύματα μήκους ενός χαρακτήρα ($l = 1$).

Απόδειξη. Για κάθε γραμμα m και $c \in C$ όπου $\mathcal{C} = \mathcal{M} =$ {Α, Β, ..., Ω}, υπάρχει μοναδικό $k = c - m \pmod{24}$ τ.ω. $E(m, k) = m + k \pmod{24} = c$. Συνεπώς, για κάθε m, c ισχύει $Pr_{k \leftarrow \mathcal{K}}[k \leftarrow \mathcal{K} : E(m, k) = c] = 1/24$, όπου πληρεί τον ορισμό του C. Shannon. □

Στην συνέχεια θα εξετασθεί για μηνύματα μήκους μεγαλύτερο από 1 ($l > 1$).

Απόδειξη. Έστω $m_1 =$ ΑΒ , $m_2 =$ ΑΩ και $c =$ ΒΓ. Τότε υπάρχει κλειδί $k \in \mathcal{K}$ τ.ω. $E(m_1, k) = c$, για $k = 1$. Ωστόσο, για κάθε $k \in K$ υπάρχει $E(m_2, k) \neq c$ και συνεπώς $Pr_{K \leftarrow \mathcal{K}}[E(m_1, K) = c] = 1/24$, όμως $Pr_{K \leftarrow \mathcal{K}}[E(m_2, K) = c] = 0$ άρα, δεν πληρείτε ο ορισμός C. Shannon. □

4.2.2 Συμπέρασμα

Από τα παραπάνω, καταλήγουμε στο ότι ένα σύστημα μετατόπισης μπορεί να έχει τέλεια ασφάλεια αν-ν το μέγεθος το μηνύματος είναι ίσο με 1.

5 One Time Pad

5.1 Εκφώνηση

Υλοποιήστε τον OTP αφού αρχικά μετατρέψετε το μήνυμα σας σε bit με χρήση του παρακάτω πίνακα. Θα πρέπει να δουλεύει η κρυπτογράφηση και η αποκρυπτογράφηση. Το μήνυμα δίνεται κανονικά και έσωτερικά μετατρέπεται σε bits. Το κλειδί είναι διαλεγμένο τυχαία και έχει μήκος όσο το μήκος του μηνύματος σας. Το αποτέλεσμα δίνεται όχι σε bits αλλά σε λατινικούς χαρακτήρες.

5.2 Λύση

5.2.1 Λεπτομέρειες για την υλοποίηση

Εκτέλεση αρχείου:

```
python otp.py <μήνυμα>
```

Τα argument <μήνυμα> είναι προαιρετικό σε περίπτωση που θέλει ο χρήστης να χρησιμοποιήσει το One Time Pad με κάποιο δικό του μήνυμα.

5.2.2 Συμπέρασμα

Δίνοντας το μήνυμα:

```
MISTAKES ARE AS SERIOUS AS THE RESULTS THEY CAUSE
```

Για κάποια εκτέλεση του κώδικα (καθώς βασίζεται στην ψευδοτυχαιότητα) έχουμε το κρυπτογραφημένο μήνυμα:

```
TR-!W!(UOYPLS)LRNLZSSF?LLXYY)HFSPTPCGOJVS
```

Όπου θα αποκρυπτογραφηθεί ως εξής:

```
MISTAKESAREASSERIOUSASTHERESULTSTHEYCAUSE
```

6 Η αριθμοθεωρητική συνάρτηση Möbius

6.1 Εκφώνηση

Αποδεικνύεται ότι το πλήθος των ανάγωγων πολυωνύμων βαθμού n στο σώμα $\mathbb{F}_2$ είναι

$$N_2(n) = \frac{1}{n} \sum_{d|n} \mu(d) \cdot 2^{n/d}, \tag{12}$$

όπου

$$\mu(d) = \begin{cases} 1 & d = 1 \\ (-1)^k & d = p_1 p_2 \cdot \cdots_k p_i : \text{πρώτοι} \\ 0 & \text{αλλού} \end{cases} \tag{13}$$

Με το σύμβολο $d|n$ εννοούμε όλους τους θετικούς διαιρέτες του n. π.χ. αν $n = 30$, τότε

$$\{d|n : 1 \leq d \leq n\} = \{1, 2, 3, 5, 6, 10, 15, 30\} \tag{14}$$

Με χρήση του συστήματος sagemath υπολογίστε το $N_2(10)$.

6.2 Λύση

6.2.1 Λεπτομέρειες για την υλοποίηση

```
python moebius.py <n>
```

Το argument <n> είναι προαιρετικό σε περίπτωση που θέλει ο χρήστης να υπολογίσει την σχέση 12 με άλλο n, οι προκαθορισμένη τιμή είναι αυτή της εκφώνησης.

Δημιουργήθηκαν τρεις βασικές συναρτήσεις, μια για τον υπολογισμό του (d), μια για τον υπολογισμό του συνόλου $d|n$ και μια για την τιμή $N_2(n)$.

6.2.2 Συμπέρασμα

Η τιμή που υπολογίσθηκε είναι $N_2(10) = 99$.

7 Rivest Cipher 4 (RC4)

7.1 Εκφώνηση

Υλοποιήστε τον RC4. Χρησιμοποιώντας το κλειδί HOUSE κρυπτογραφήστε το μήνυμα (ξαναγράψτε το χωρίς κενά):

```
MISTAKES ARE AS SERIOUS AS THE RESULTS THEY CAUSE
```

Η υλοποίηση σας πρέπει και να αποκρυπτογραφεί σωστά.

7.2 Λύση

7.2.1 Λεπτομέρειες για την υλοποίηση

Εκτέλεση αρχείου:

```
python rc4.py <μήνυμα> <κλειδί>
```

Τα arguments <μήνυμα> και <κλειδί> είναι προαιρετικά σε περίπτωση που θέλει ο χρήστης να το RC4 με άλλες τιμές, οι προκαθορισμένες τιμές είναι αυτές που δίνονται στην εκφώνηση.

7.2.2 Συμπέρασμα

Εκτελώντας το αρχείο για τις τιμές της εκφώνησης, επιστρέφεται το εξής κρυπτογραφημένο μήνυμα:

```
IGD!APO-TJUQPDSPMAOZUIAZ(VF(VFGQ.IIWMB(WX
```

Ενώ η αποκρυπτογραφημένη μορφή του παραπάνω μηνύματος είναι:

```
MISTAKESAREASSERIOUSASTHERESULTSTHEYCAUSE
```

8 Διαφορική ομοιομορφία

8.1 Εκφώνηση

Αν Σ ένα σύνολο με |Σ| συμβολίζουμε το πλήθος των στοιχείων του, υπολογίστε τη διαφορική ομοιομορφία (differential uniformity) του S-box,

$$Diff(S) = \max_{x \in \{0,1\}^6 - \{0\}, y \in \{0,1\}^4} |\{z \in \{0,1\}^6 : S(z \oplus x) \oplus S(z) = y\}| \quad (15)$$

Γενικά, για S-boxes:

$$S : \{0,1\}^n \to \{0,1\}^m \quad (16)$$

ο προηγούμενος ορισμός γράφεται:

$$Diff(S) = \max_{x \in \{0,1\}^n - \{0\}, y \in \{0,1\}^m} |\{z \in \{0,1\}^n : S(z \oplus x) \oplus S(z) = y\}| \quad (17)$$

και ισχύει:

$$Diff(S) \geq \max\{2, 2^{n-m}\} \quad (18)$$

Όσο μικρότερη είναι αυτή η ποσότητα, τόσο πιο ανθεκτικό είναι το S-box στη διαφορική κρυπτανάλυση.

8.2 Λύση

8.2.1 Λεπτομέρειες για την υλοποίηση

Βιβλιοθήκες

- matplotlib
- matplotlib.pyplot

Εκτέλεση αρχείου:

```
python sbox.py
```

Το αρχείο, περιέχει ουσιαστικά την υλοποίηση της σχέσης (15). Η υλοποίηση έγινε με βάση το βιβλίο [3] που χρησιμοποιηθεί την συνάρτηση Feistel που περιγράφεται μέσα.

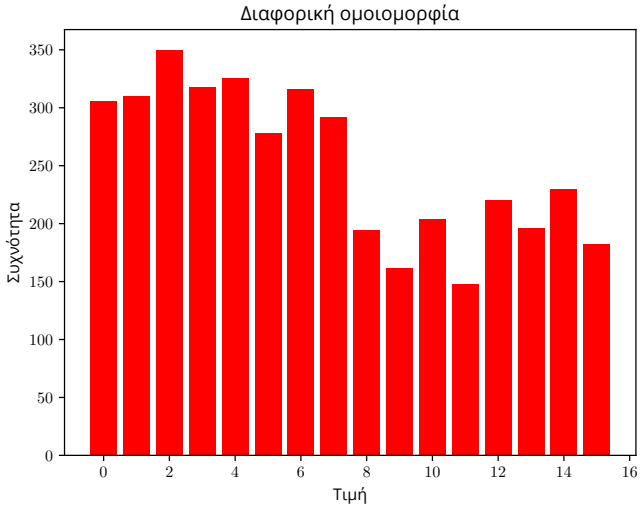
8.2.2 Συμπέρασμα

Για το S-box της εκφώνησης ισχύει: $n = 6, m = 4$. Έτσι η σχέση (18) έχει ως εξής:

$$Diff(S) \geq \max\{2, 2^{n-m}\} = \max\{2, 2^{6-4}\} = \max\{2, 4\} = 4 \Rightarrow Diff(S) \geq 4 \tag{19}$$

Δηλαδή, αναμένεται η διαφορική ομοιομορφία να έχει τιμή τουλάχιστον ίση με 4.

Εκτελώντας τον κώδικα, παράγεται το εξής γράφημα:



Σχήμα 2: Διαφορική ομοιομορφία του S-box (15)

Όπως φαίνεται λοιπόν από το σχήμα 2 η μέγιστη συχνότητα εμφάνισης είναι το 350 για όλες τι πιθανές τιμές που μπορεί να πάρει το x (πλήθος τιμών του x είναι $2^6 - 1 = 63$), δηλαδή $Diff(S) = \max|\{S(z \oplus x) \oplus S(z)\}| = 350$.

9 Avalanche effect

9.1 Εκφώνηση

Εξετάστε αν ισχύει το avalanche effect στο AES-128. Αναλυτικότερα, φτιάξτε αρκετά ζευγάρια (≥ 30) μηνυμάτων (m_1, m_2) που να διαφέρουν σ ένα bit. Εξετάστε σε πόσα bits διαφέρουν τα αντίστοιχα κρυπτομηνύματα. Δοκιμάστε με δύο καταστάσεις λειτουργίας: ECB και CBC (η δεύτερη θέλει

και IV block). Τα μήκη των μηνυμάτων που θα χρησιμοποιήσετε να έχουν μήκος διπλάσιο του μήκους ενός block. Δηλαδή για τον AES-128, να είναι μήκους 256-bits.

9.2 Λύση

9.2.1 Λεπτομέρειες για την υλοποίηση

Βιβλιοθήκες

- Crypto.Cipher
- bitstring
- statistics

Εκτέλεση αρχείου:

```
python avalanche.py
```

Δοκιμάσθηκαν 200 τυχαία ζευγάρια δυαδικών μηνυμάτων, χρησιμοποιώντας τυχαία δυαδικών κλειδιά μήκους 128-bit αντίστοιχα και για το IV. Για την χρήση του AES, έγινε χρήση της βιβλιοθήκης Crypto όπου εξετάσθηκαν και οι δυο λειτουργίες.

9.2.2 Συμπέρασμα

Έπειτα από αρκετές προσομοιώσεις, οι μέσοι όροι διαφορών στα ζεύγη των μηνυμάτων έχουν ως εξής: για ECB 64-bit και για CBC 128-bit. Αυτό σημαίνει πως στην λειτουργία ECB η αλλαγή ενός bit επιφέρει αλλαγή στο $\approx 25\%$ του μηνύματος άρα, δεν παρουσιάζεται το avalanche effect σε αυτή την λειτουργία. Αντιθέτως, στην λειτουργία CBC αλλάζει το $\approx 50\%$ του μηνύματος που σημαίνει ότι υπάρχει το avalanche effect.

10 Capture The Flag

10.1 Εκφώνηση

Μπορείτε να ανοίξετε το secure.zip?

10.2 Λύση

```
be121740bf988b2225a313fa1f107ca1
```

10.3 Πορεία

- `aHR0cHM6Ly9jcnlwdG9sb2d5LmNzZC5hdXRoLmdyOjgwODAvaG9tZS9wdWIvMTUv`
- `https://cryptology.csd.auth.gr:8080/home/pub/15/`
- `#heretic!`
- `https://tinyurl.com/26ru4359`
- `f1f5e44313a1b684f1f7e8eddec4fcb0`

Αναφορές

[1] The Mad Doctor, *Five Ways to Crack a Vigenère Cipher*, University of Southampton, 2020.

[2] Stanislaw Jarecki, *Lecture 1: Crypto Overview, Perfect Secrecy, One-time Pad*, Donald Bren School of Information and Computer Sciences, 2004.

[3] Δραζιώτης, Κ., *Εισαγωγή στην Κρυπτογραφία* [Προπτυχιακό εγχειρίδιο], Κάλλιπος, Ανοικτές Ακαδημαϊκές Εκδόσεις, 2022.