

POLITECHNIKA ŚWIĘTOKRZYSKA

LABORATORIUM CYBERBEZPIECZEŃSTWO

Numer ćwiczenia:

5

Temat ćwiczenia:

WEP/WPA2 PSK/WPA2 RADIUS

Damian Zdyb

Data wykonania:

13.12.2018

Data oddania do sprawdzenia:

15.12.2018

Ocena:

1.Wstęp:

Po podzieleniu się w grupy 4osobowe przystąpiliśmy do laboratoriów, Scenariusz na bieżąco przedstawiał nam Prowadzący.

Celem Zadania było:

- Podzielenie się na 2podzespoły.
- Pierwszy zespół (2osoby) zajmowały się analizą kodu
- Drugi zespół (również 2osoby) zajmowały się atakiem BruteForce

2. Przebieg ćwiczenia:

Do wykonania zadania niezbędne były pliki Attack.py oraz Cipher.py .

W pierwszym jest kod, który łamie hasła zaszyfrowane metodą częstotliwościową, używając algorytmu BruteForce :

```
message = 'XOY WBTCFAIXS DCFHOZ AWSRNWCKS.DZ, K BCQM N  
DWAHYI BO GCPCHS RCGNLC RC WBBSUC KGHFNAGI C GWZS  
HNK.UÓFBWQNSX GNÓGHYW. K FSXCBWS NOUFCZSBWO  
NBOXRCKOŁ GWĘ XSR SB UÓFBWY. CDSFOHCF ŁORCKOFYW NCGHOŁ  
IKWENWCBM K YOPWBWS AOGNMBM. - DC UCRNWBBSX OYQXW  
FOHCKBWQM IKCZBWZW DFOQCKBWYO, BWQ AI GWĘ BWS GHOLC.  
OYQXO NOYCŃQNMŁO GWĘ GNQNEŚZWKWS - AÓKW DCFHOZCKW  
OBBO CGORQNIY. RNWSŃ KQNSŚBWSX NWSAWO NOHFNEĞŁO GWĘ K  
WBBMA FSXCBWS YCDOZBW FIRBO. HFNSQV UÓFBWYÓK NCGHOŁC  
FOBBMQVRKO ZOHO HSAI HOYŻS RCGNC RC KGHFNARI K YCDOZBW  
K DCZYCKWQOQV. CŚAWI UÓFBWYÓK NUWBĘLC K KMBWYI NOKOŁI  
K YCDOZBW FIRBO. DFNSN YWZYORNWSGWĄH UCRNWB HFKOŁO  
OYQXO FOHCKBWQNO.'
```

```
LETTERS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
```

```
for key in range(len(LETTERS)):
```

```
    translated = ''
```

```
    for symbol in message:
```

```
        if symbol in LETTERS:
```

```
            num = LETTERS.find(symbol)
```

```
            num = num - key
```

```
            if num < 0:
```

```
                num = num + len(LETTERS)
```

```
            translated = translated + LETTERS[num]
```

```
        print
```

```
    else:
```

```
        translated = translated + symbol
```

```
print('Key #s: %s' % (key, translated))
```

Na początku dostarczamy skryptowi zaszyfrowany tekst (message).

Następnie dla każdej litery przypisuje cyfrę, oraz odejmuje od danego numeru Key, czyli o ile został przesunięty ciąg znaków.

Dla porównania: Key 10 dla message:

Key #10: NEO MRJSVQYNI TSVXEP QMIHDMSAI.TP, A RSGC D TMȦXOY
RE WSFSXI HSWDŁS HS MRRIKS A WXVDȦWY S WMPI
XDA.KÓVRMGDIN WDÓWXOM. A VINSRMI DEKVṠZIRME DRENHSAEŁ
WMĘ NIHIR KÓVRMO. STIVEXSV ŁEHSAEVOM DSWXEŁ YAMĘDMSRC
A OEFMRMI QEWDRC. - TS KSHDMRRIN EOGNM VEXSARMGC
YASPRMPM TVEGSARMOE, RMG QY WMĘ RMI WXEŁS. EOGNE
DEOṠNGDCŁE WMĘ WDGDĖSPMAMI - QÓAM TSVXEPSAM ERRE
SWEHGDYO. HDMIŃ AGDİSRMIN DMIQME DEXVDĘWŁE WMĘ A
MRRCQ VINSRMI OSTEPRM VYHRE. XVDIGL KÓVRMOÓA DSWXEŁS
VERRCGLHAE PEXE XIQY XEȮZI HSWDS HS A WXVDȦHY A OSTEPRM
A TSPOSAMGEGL. ṠSQMY KÓVRMOÓA DKMRĘŁS A ACRMOY DEAEŁY
A OSTEPRM VYHRE. TVDID OMPOEHDMIWMȦX KSHDMR XVAEŁE
EOGNE VEXSARMGDE.

Oraz Key 14 (prawidłowy) :

Key #14: JAK INFORMUJE PORTAL MIEDZIOWE.PL, W NOCY Z PIĄTKU
NA SOBOTE DOSZŁO DO INNEGO WSTRZĄSU O SIŁE TZW.GÓRNICZEJ
SZÓSTKI. W REJONIE ZAGROŻENIA ZNAJDOWAŁ SIĘ JEDEN GÓRNIK.
OPERATOR ŁADOWARKI ZOSTAŁ UWIEŻZIONY W KABINIE MASZYN. -
PO GODZINNEJ AKCJI RATOWNICY UWOLNILI PRACOWNIKA, NIC MU
SIĘ NIE STAŁO. AKCJA ZAKOŃCZYŁA SIĘ SZCZĘŚLIWIE - MÓWI
PORTALOWI ANNA OSADCZUK. DZIEŃ WCZEŚNIEJ ZIEMIA
ZATRZĘSŁA SIĘ W INNYM REJONIE KOPALNI RUDNA. TRZECH
GÓRNIKÓW ZOSTAŁO RANNYCH DWA LATA TEMU TAKŻE DOSZO DO
WSTRZĄDU W KOPALNI W POLKOWICACH. OŚMIU GÓRNIKÓW
ZGINĘŁO W WYNIKU ZAWAŁU W KOPALNI RUDNA. PRZEZ
KILKADZIESIĄT GODZIN TRWAŁA AKCJA RATOWNICZA.

Plik Cipher.py zawiera kod dla szyfrowania :

```
message = ' ' #nasza wiadomosc
```

```
key = 14 #Clue
```

```
mode = 'encrypt'
```

```
LETTERS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ' # Clue
```

```
# plaintext -> ciphertext or reversed
```

```
translated = "
```

```
# capitalize the string in message
```

```
message = message.upper()
```

```
for symbol in message:
```

```
if symbol in LETTERS:
```

```
# get the encrypted (or decrypted) number for this symbol
```

```
num = LETTERS.find(symbol) # get the number of the symbol
```

```
if mode == 'encrypt':
```

```
num = num + key
```

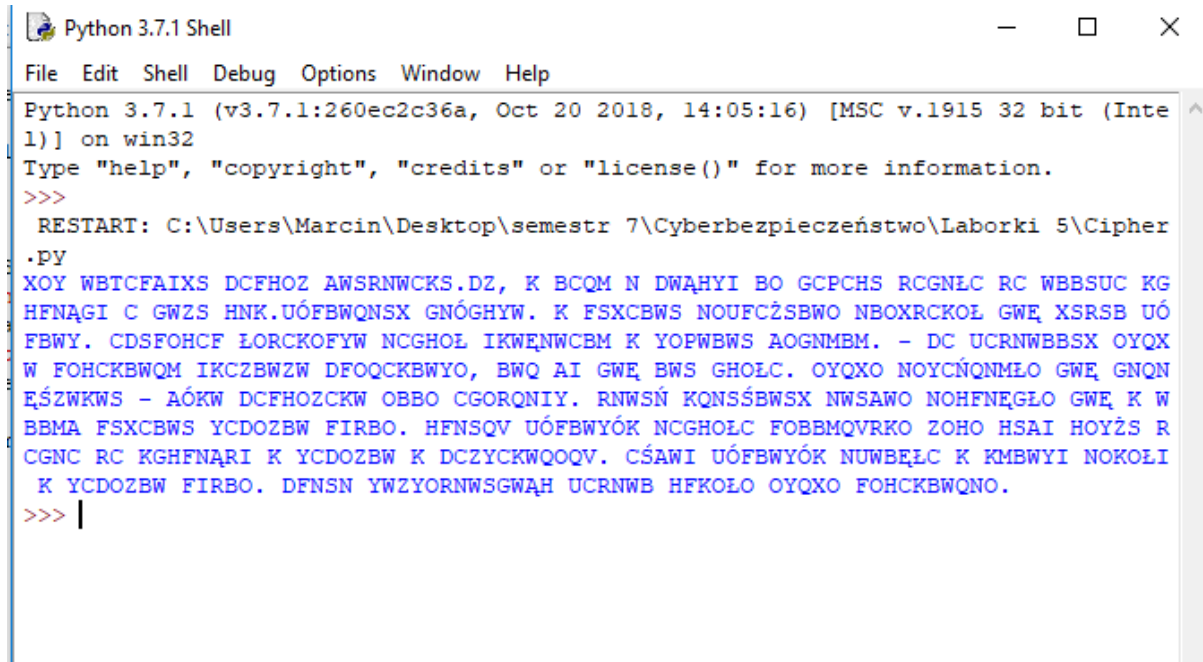
```
elif mode == 'decrypt':
```

```
num = num - key
```

```
# wrap-around if num > length of LETTERS or less than 0
```

```
if num >= len(LETTERS):  
    num = num - len(LETTERS)  
elif num < 0:  
    num = num + len(LETTERS)  
# add encrypted/decrypted number's symbol at the end of translated  
translated = translated + LETTERS[num]  
else:  
    # just add the symbol without encrypting/decrypting  
    translated = translated + symbol  
  
print(translated)
```

Powyższy kod szyfruje naszą wiadomość. Po uruchomieniu go z odpowiednim message otrzymamy:



```
Python 3.7.1 Shell
File Edit Shell Debug Options Window Help
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\Marcin\Desktop\semestr 7\Cyberbezpieczeństwo\Laborki 5\Cipher
.py
XOY WBTCFAIXS DCFHOZ AWSRNWCKS.DZ, K BCQM N DWĄHYI BO GCPCHS RCGNLC RC WBBSUC KG
HFNAGI C GWZS HNK.UÓFBWQNSX GNÓGHYW. K FSXCBWS NOUFCZSBWO NBOXRCKOŁ GWĘ XSRSB UÓ
FBWY. CDSFOHCF ŁORCKOFYW NCGHOŁ IKWĘNWCBM K YOPWBWS AOGNMBM. - DC UCRNWBSX OYQX
W FOHCKBWQM IKCZBWZW DFOQCKBWYO, BWQ AI GWĘ BWS GHOLC. OYQXO NOYCŃQNMŁO GWĘ GNQN
ĘŚZWKWS - AÓKW DCFHOZCKW OBBO CGORQNIY. RNWSŃ KQNSŚBWSX NWSAWO NOHFNEGŁO GWĘ K W
BBMA FSXCBWS YCDOZBW FIRBO. HFNSQV UÓFBWYÓK NCGHOŁC FOBBMQVRKO ZOHO HSAI HOYŻS R
CGNC RC KGHFNARI K YCDOZBW K DCZYCKWQOQV. CŚAWI UÓFBWYÓK NUWBĘŁC K KMBWYI NOKOŁI
K YCDOZBW FIRBO. DFNSN YWZYORNWSGWĄH UCRNWB HFKOŁO OYQXO FOHCKBWQNO.
>>> |
```

Na potrzebę ćwiczenia dodaliśmy pętlę for, która zlicza nam ilość literek.

```
for char in LETTERS:
```

```
    count = message.count(char)
```

```
    if count !=0 :
```

```
        print (char, count)
```

```
RESTART: C:\Users\Marcin\Desktop\semestr 7\Cyberbezpieczeństwo\Laborki
5\Cipher.py
```

A 50

B 2

C 17

D 17

E 28

F 1

G 9

H 3

I 52

J 12

K 22

L 13

M 9

N 40

O 44

P 12

R 25

S 21

T 21

U 14

W 30

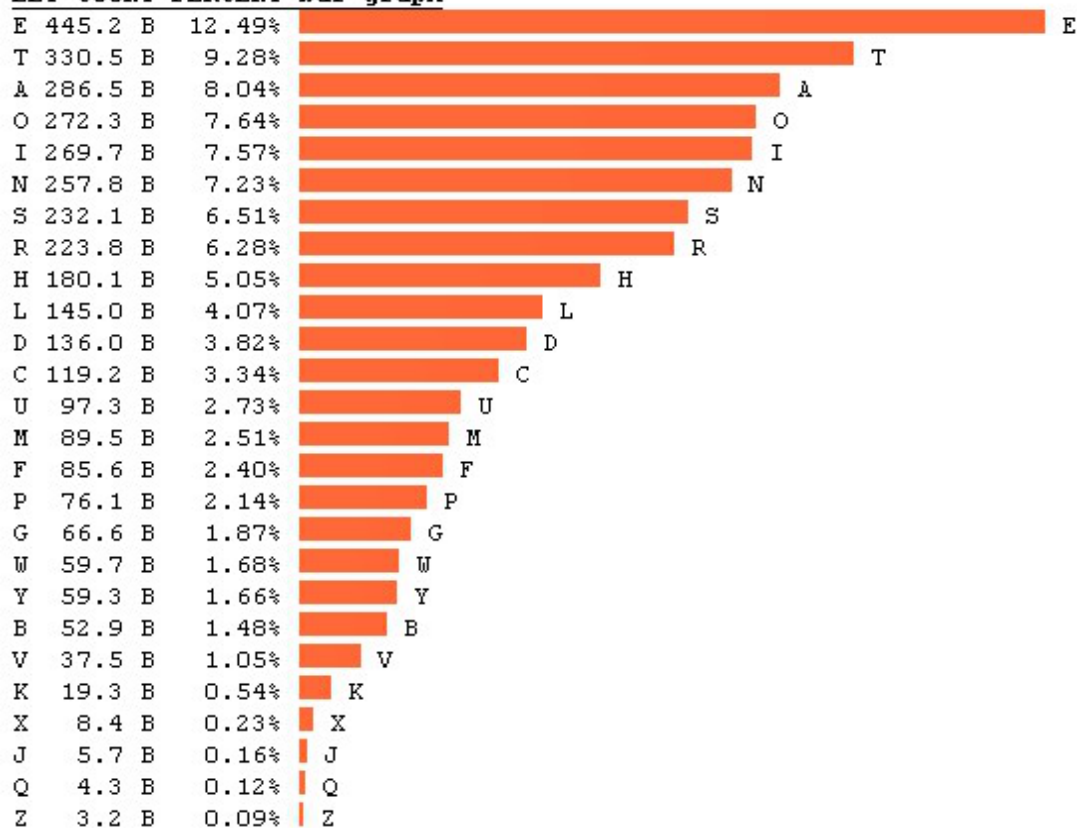
Y 9

Z 34

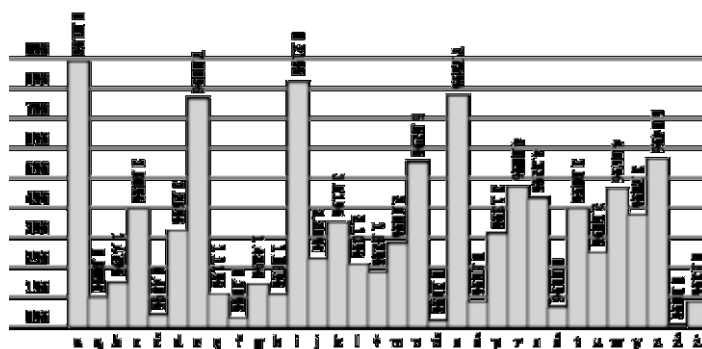
W ten sposób sprawdziliśmy , która litera pojawia się najczęściej oraz skorzystaliśmy z diagramu częstotliwości użycia liter, aby sprawdzić najbardziej prawdopodobną literę.

Dla języka angielskiego:

LET COUNT PERCENT bar graph



Dla języka polskiego:

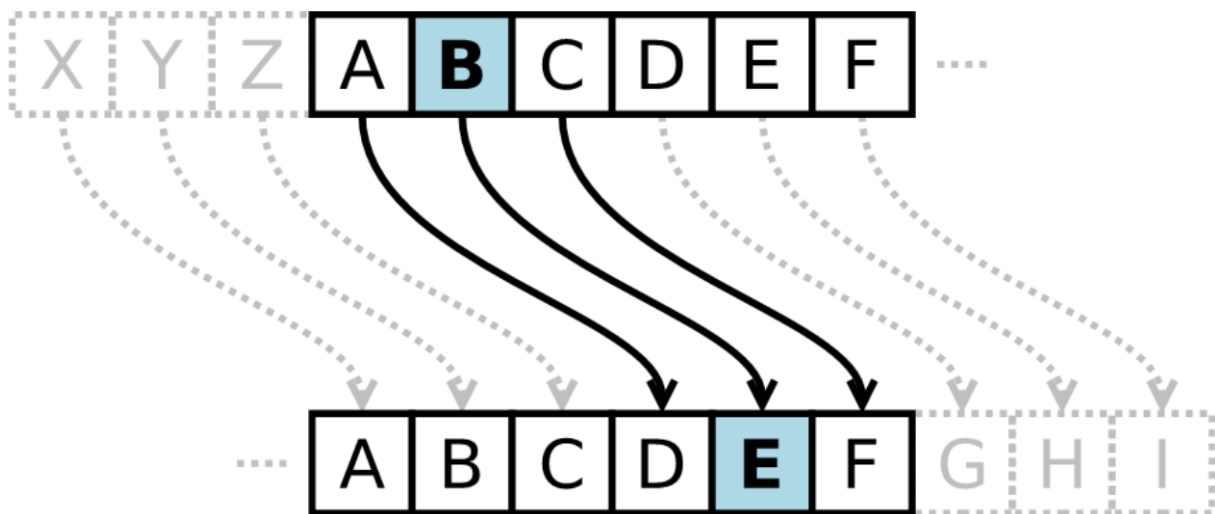


Treść artykułu była w języku Polskim, więc wg diagramu litera A jest najczęściej powtarzającym się znakiem.

Litera 'a' leży w odległości 14znaków od litery 'o' , więc nasz klucz wynosi 14.

Dodając klucz do każdej zakodowanej litery otrzymamy naszą wiadomość.

Jest to tak zwany KOD CEZARA.



Aby ulepszyć szyfr Cezara należy wg mnie wykonać szyfrowanie podwójne, z dwoma różnymi kluczami.

Szyfr Vigenère’a jest natomiast szyfrem Cezara ze zmiennym przesunięciem na każdej pozycji w tekście. Wartość przesunięcia jest definiowana przez dowolne słowo kluczowe. Jeśli słowo kluczowe jest losowe i o długości nie krótszej niż sama wiadomość, wtedy jest to szyfr z kluczem jednorazowym, niemożliwy do złamania, pod warunkiem utrzymania klucza w tajemnicy. Klucz krótszy od wiadomości niesie ze sobą powtarzający się wzór, który może być rozpoznany przez zaawansowane techniki analizy częstościowej

Confusion and Defusion

Confusion oznacza, że każda cyfra binarna (bit) tekstu zaszyfrowanego powinna zależeć od kilku części klucza, przesłaniając połączenia między tymi dwoma.

Defusion oznacza, że jeśli zmienimy pojedynczy bit zwykłego tekstu, to (statystycznie) połowa bitów w zaszyfrowanym tekście powinna się zmienić, i podobnie, jeśli zmienimy jeden bit zaszyfrowanego tekstu, wówczas powinna zmienić się około połowa bitów tekstowych. Ponieważ bit może mieć tylko dwa stany, kiedy wszystkie zostaną ponownie ocenione i zmienione z jednej pozornie losowej pozycji na drugą, połowa bitów zmieni się.