Assignment 2

Maxime CHAMBREUIL
maxime.chambreuil@mail.mcgill.ca

Contents

1 Exercise 1

1.1 Substitution cypher

1.2 Viegenere cypher

1.3 Affine cypher

KQEREJEBCPPCJCRKIEACUZBKRVPKRBCIBQCARBJCVFCUP
KRIOFKPACUZQEPBKRXPEIIEABDKPBCPFCDCCAFIEABDKP
BCPFEQPKAZBKRHAIBKAPCCIBURCCDKDCCJCIDFUIXPAFF
ERBICZDFKABICBBENEFCUPJCVKABPCYDCCDPKBCOCPERK
IVKSCPICBRKIJPKABI

If you calculate the frequencies of letters, you will find that C (2) and B (1) appeared at most. They stand for E (4) and T (19) so we have the system :

$$\begin{cases} 2a + b = 4 \\ a + b = 19 \end{cases} \Rightarrow \begin{cases} a = 11 \\ b = 8 \end{cases}$$

O CANADA TERRE DE NOS AIEUX TON FRONT EST CEINT DE FLEURONS GLORIEUX CAR TON BRAS SAIT PORTER L EPEE IL SAIT PORTER LA CROIX TON HISTOIRE EST UNE EPOPEE DES PLUS BRILLANTS EXPLOITS ET TA VALEUR DE FOI TREMPEE PROTEGERA NOS FOYERS ET NOS DROITS

1.4 Unspecified cypher

2 Exercise 2

2.1 Property

As we have studied in class, we know that :

$$H \text{ is Strongly Universal}_2 \iff Pr[h(a) = b] = \frac{1}{|B|}$$
$$\text{and } Pr[h(a_1) = b_1; h(a_2) = b_2] = \frac{1}{|B|^2}$$
$$\text{and } Pr[h(a_1) = b_1 / h(a_2) = b_2] = \frac{1}{|B|}$$

2.2 Unicity of h(a)

Suppose we have a_1 and a_2 such that $h(a_1) = h(a_2)$, so

$$a_1 M \oplus Y = a_2 M \oplus Y$$
$$a_1 M = a_2 M$$

$a_1 = a_2$ iff M is invertible.

If M is not invertible, H is not a bijection and we can have many favorable cases to the fact that h(a) = b. So we can say that H_0 is not Strongly Universal$_2$.

Concerning H_1, we have 2^m elements in B and one favorable case, so :

$$Pr[h(a) = b] = \frac{1}{2^m} = \frac{1}{|B|}$$

Events are independent, so :

$$Pr[h(a_1) = b_1; h(a_2) = b_2] = \frac{1}{2^m} \times \frac{1}{2^m} = \frac{1}{|B|^2}$$

$$Pr[h(a_1) = b_1 / h(a_2) = b_2] = \frac{Pr[h(a_1) = b_1; h(a_2) = b_2]}{Pr[h(a) = b]} = \frac{\frac{1}{|B|^2}}{\frac{1}{|B|}} = \frac{1}{|B|}$$

3 Exercise 3

3.1 Finding the polynom P

```
getPoly := proc(deg,field);
        myPolynom;
        myPolynom := RandomTools[Generate](polynom(integer(range=0..1),
                                                    x,degree=deg)):
        while (not (Irreduc(myPolynom) mod 2)) do
                myPolynom := RandomTools[Generate](polynom(
```

```
                                integer(range=0..field-1),x,degree=deg)):
        end do;
        myPolynom;
end proc;

pol := getPoly(1000,2);
```

3.2 Building the field F

```
F2 := GF(2,1000,pol);
```

3.3 Finding the primitive element g

```
g := F2[PrimitiveElement](x);
```

3.4 Picking up 2 random elements i and j

```
i := F2[random](x);
j := F2[random](x);
```

3.5 Telling x and y

```
x:=1:
y:=1:

while (F2['^'](g,x) <> i and ( x >= 2^1000-1)) do:
       x := x + 1:
end do:

while (F2['^'](g,y) <> j and (y >= 2^1000-1)) do:
       y := y + 1:
end do:
```

3.6 Picking up a message m and calculating the tag t

```
m := RandomTools[Generate](polynom(integer(range=0..1),z,degree=1000));

tmp := F2['*'](m,i):
t := F2['+'](tmp,j):

for k from 0 by 1 to 49 do:
       t[k] := coeff(tmp,z,k);
end do:
```

4 Exercise 4

4.1 Decrypting the cyphertext

```
with(linalg):

decrypt := proc(K,cypher,modulo):

M := inverse(K);
```

```
message:
tmp:
i := 1:

tmp := Matrix(cypher[1,i..i+3]):
tmp := tmp.M:
message := seq (tmp[1,j+1] mod modulo ,j=0..3):

i := i + 4:

while (i < 28) do:
tmp := Matrix(cypher[1,i..i+3]):
tmp := tmp.M:
message := seq (tmp[1,j+1] mod modulo ,j=0..3):
i := i + 4:
end do:
message:
end proc:


K := Matrix([[1,2,3,4],[2,3,4,0],[3,4,0,0],[4,0,0,0]]):
c := Matrix([23,06, 16, 08, 12, 10, 26, 18, 20, 21, 13 ,14 ,
  22 ,04, 27, 18 ,25, 07, 06, 24, 21, 20, 16 ,18, 17, 08, 02, 23]):

plain := decrypt(K,c,29);
```