The Sage Notebook
Version 5.13

admin Toggle | Home | Published | Log | Settings | Help | Report a Problem | Sign out

TP5

last edited Feb 11, 2014, 8:16:33 AM by michel.cretin

File... Action... Data... sage Typeset

Print Worksheet Edit Text Revisions Share Publish

Save Save & quit Discard & quit

```
restart
```

```
p = 3; A = PolynomialRing(GF(p),'X'); X = A.gen(); A
```

```
Univariate Polynomial Ring in X over Finite Field of size 3
```

```
P = X^5+X^4+X^2+1; P
```

```
X^5 + X^4 + X^2 + 1
```

```
m = P.degree(); m
```

```
5
```

```
K = GF(p^m,name='alpha',modulus=P); alpha = K.gen(); K
```

```
Finite Field in alpha of size 3^5
```

```
n = alpha.multiplicative_order()
n
```

```
242
```

```
delta = 8
delta
```

```
8
```

```
g =X^25 + X^24 + X^22 + X^21 + X^20 + 2*X^18 + 2*X^17 + 2*X^16 + 2*X^15 + X^13 + 2*X^12 + X^10 + 2*X^8 + X^4 + X^3 + X^2 + 2*X + 1
g
```

```
X^25 + X^24 + X^22 + X^21 + X^20 + 2*X^18 + 2*X^17 + 2*X^16 + 2*X^15 +
X^13 + 2*X^12 + X^10 + 2*X^8 + X^4 + X^3 + X^2 + 2*X + 1
```

```
k = n -g.degree()
k
```

```
217
```

```
fr=X^224+X^110+X^109+X^107+X^106+X^105+2*X^103+X^102+X^101+2*X^100+2*X^99+X^97+2*X^95+X^94+X^92+2*X^90+2*X^89+X^88+2*X^86+2*X^85+X^83+X^82+2*X^81+2*X^78+2*X^77+2*X^76
+3*X^75+2*X^74+2*X^73+X^71+2*X^70+X^68+2*X^66+2*X^64+2*X^63+X^62+X^59+X^58+X^57+X^56+X^55+X^54+2*X^52+X^51+2*X^49+X^47+2*X^43+2*X^42+2*X^41+X^40+2*X^39+2*X^19
fr
```

```
X^224 + X^110 + X^109 + X^107 + X^106 + X^105 + 2*X^103 + X^102 + X^101
+ 2*X^100 + 2*X^99 + X^97 + 2*X^95 + X^94 + X^92 + 2*X^90 + 2*X^89 +
X^88 + 2*X^86 + 2*X^85 + X^83 + X^82 + 2*X^81 + 2*X^78 + 2*X^77 + 2*X^76
+ 2*X^74 + 2*X^73 + X^71 + 2*X^70 + X^68 + 2*X^66 + 2*X^64 + 2*X^63 +
X^62 + X^59 + X^58 + X^57 + X^56 + X^55 + X^54 + 2*X^52 + X^51 + 2*X^49
+ X^47 + 2*X^43 + 2*X^42 + 2*X^41 + X^40 + 2*X^39 + 2*X^19
```

fr n'est pas un mot du code puisque non divisible par g

```
Q,R = fr.quo_rem(g)
R <> 0
```

```
True
```

le syndrome

```
S = add([fr(X=alpha^i)*X^(i-1) for i in range(1,delta)])
show(S)
```

$$\left(\alpha^{4} + 2\alpha^{2} + \alpha\right)X^{6} + \left(2\alpha^{4} + 2\alpha^{3} + 2\alpha^{2} + 2\alpha\right)X^{5} + \left(2\alpha^{4} + 2\alpha^{3} + \alpha + 2\right)X^{4} + \left(2\alpha^{2} + \alpha\right)X^{3} + \left(2\alpha^{4} + 2\alpha^{3} + 2\alpha^{2} + 2\alpha + 1\right)X^{2} + \left(2\alpha^{4} + \alpha^{3} + 2\alpha^{2} + 2\alpha + 1\right)X + 2\alpha^{4} + \alpha^{3} + 2\alpha^{2} + 2\alpha + 2$$

algorithme d'Euclide-Sujuyama

```
R0 = X^(delta-1)
R1 = S
U0 = 1
U1 = 0
V0 = 0
V1 = 1
while True:
    Q,R2 = R0.quo_rem(R1)
    U2 = U0 -Q*U1
    V2 = V0 -Q*V1
    if R2.degree() < (delta-1)/2:
        break
    R0,R1 = R1,R2
    U0,U1 = U1,U2
    V0,V1 = V1,V2
```

```
R = R2
show(R)
```

$$\left(\alpha^{4} + 2\alpha\right)X^{2} + \left(2\alpha^{4} + 2\alpha + 2\right)X + 2\alpha^{2} + 1$$

```
V = V2
show(V)
```

$$\left(2\alpha^{4} + \alpha\right)X^{3} + \left(\alpha^{4} + 2\alpha^{3} + 2\alpha^{2}\right)X^{2} + \left(2\alpha^{4} + 2\alpha^{3} + \alpha^{2} + 2\alpha + 2\right)X + 2\alpha^{3} + \alpha^{2}$$

```
c = V(X=0)
show(c)
```

$$2\alpha^{3} + \alpha^{2}$$

le polynôme localisateur d'erreurs

```
sigma = V/c
show(sigma)
```

$$\left(2\alpha^4 + 2\alpha^3 + 1\right)X^3 + \left(2\alpha^4 + \alpha^3 + \alpha^2 + 2\alpha\right)X^2 + \left(2\alpha^4 + \alpha^2\right)X + 1$$

le nombre d'erreurs

```
t = sigma.degree()
t
```

```
3
```

le polynôme évaluateur d'erreurs

```
omega= R/c
show(omega)
```

$$\left(\alpha^4 + \alpha^3 + 2\right)X^2 + \left(2\alpha^4 + \alpha^2 + 2\right)X + 2\alpha^4 + \alpha^3 + 2\alpha^2 + 2\alpha + 2$$

les racines de σ

```
rc =map(lambda e : e[0] ,sigma.roots())
show(rc)
```

$$\left[\alpha^3 + 2\alpha + 1, 2\alpha^4 + \alpha^3, \alpha^4 + 2\alpha^3 + \alpha^2 + \alpha + 2\right]$$

la position des erreurs

```
d = map(lambda e : n-e, map(lambda r: r.log(alpha),rc))
d
```

```
[75, 19, 224]
```

la formule de Forney

```
c = [-omega(X=r)/sigma.derivative()(X=r) for r in rc]
c
```

```
[1, 2, 1]
```

l'eereur

```
e =add([c[i]*X^d[i] for i in range(0,t)])
e
```

```
X^224 + X^75 + 2*X^19
```

le mot codé corrigé

```
f = fr-e
f
```

```
X^110 + X^109 + X^107 + X^106 + X^105 + 2*X^103 + X^102 + X^101 +
2*X^100 + 2*X^99 + X^97 + 2*X^95 + X^94 + X^92 + 2*X^90 + 2*X^89 + X^88
+ 2*X^86 + 2*X^85 + X^83 + X^82 + 2*X^81 + 2*X^78 + 2*X^77 + 2*X^76 +
2*X^75 + 2*X^74 + 2*X^73 + X^71 + 2*X^70 + X^68 + 2*X^66 + 2*X^64 +
2*X^63 + X^62 + X^59 + X^58 + X^57 + X^56 + X^55 + X^54 + 2*X^52 + X^51
+ 2*X^49 + X^47 + 2*X^43 + 2*X^42 + 2*X^41 + X^40 + 2*X^39
```

on vérifie que c'est bien un mot du code et on trouve le mot initial h

```
h,R = f.quo_rem(g)
R
```

```
0
```

```
h
```

```
X^85 + 2*X^77 + X^58 + 2*X^39
```

evaluate