FCSC 2021 : **Lost Curve** (Write-up)

Jakobus

Mai 2021

- **Catégorie** : Crypto (toujours la meilleure)
- **Points** : 200 (score dynamique)
- **Résolutions** : 33
- **Description** : *J'ai perdu l'équation de ma courbe elliptique : pouvez-vous m'aider à la retrouver ?*
- **Fichier** : lost_curve.py
- **shell online** : nc challenges1.france-cybersecurity-challenge.fr 6002

Le script Python dont on dispose (lost_curve.py) correspond à ce à quoi on accède en ligne. Le challenge consiste à retrouver, à partir de deux points P et Q d'une courbe elliptique qui nous sont donnés, les paramètres a, b, p de la courbe elliptique $E : y^2 = x^3 + ax + b \pmod p$ auxquels appartiennent P et Q. Au vu du programme, on a $Q = 2P$.

Pour commencer, on considère les formules du double $Q(x_q, y_q)$ d'un point $P(x_p, y_p)$:

$$\lambda = \frac{3x_p^2 + a}{2y_p}$$
$$x_q = \lambda^2 - 2x_p$$
$$y_q = \lambda(x_p - x_q) - y_p$$

En multipliant le tout par $(2y_p)^2$, réarrangeant, réinjectant et utilisant le fait que $y_p^2 = x_p^3 + ax_p + b$ (et de même pour Q), on obtient que

$$M := y_p^2 - y_q^2 - 2y_p(y_p + y_q) + (x_q - x_p)(x_q^2 + x_p x_q - 2x_p^2) \equiv 0 \pmod p$$

On peut donc, à l'aide de factordb par exemple, déterminer un facteur premier de M susceptible d'être p (généralement ceci est assez facile, puisqu'il y a peu de chance que M contienne beaucoup de facteurs premiers de la même taille que p qui est de 80 bits. Sinon il suffit de recommencer !).
Nous obtenons donc p. On peut alors retrouver $\lambda = \dfrac{y_p + y_q}{x_p - x_q}$ puis $a = 2\lambda y_p - 3x_p^2$, puis $b = y_p^2 - x_p^3 - ax_p$ (les calculs se faisant $\pmod p$). On a à présent tous les paramètres en main ! Il ne reste plus qu'utiliser le script Sage suivant :

```
xp = Integer(input("xp : "))
yp = Integer(input("yp : "))
xq = Integer(input("xq : "))
yq = Integer(input("yq : "))

M = yp^2 - yq^2 - 2*yp*(yp + yq) + (xq - xp)*(xq^2+xp*xq -2*xp^2)

print(f"M = {M}")

p = Integer(input("p : "))

```

```
R = GF(p)

xp = R(xp)
yp = R(yp)
xq = R(xq)
yq = R(yq)

l = (yp+yq)/(xp-xq)

a = 2*l*yp - 3*xp^2

b = yp^2 - xp^3 - a*xp


print(f"a : {a}")
print(f"b : {b}")
print(f"p : {p}")
```

et obtenir le magnifique flag :)