Lost Curve (Cryptographie - 200 points)

Énoncé

J'ai perdu l'équation de ma courbe elliptique : pouvez-vous m'aider à la retrouver ?

`nc challenges1.france-cybersecurity-challenge.fr 6002`

Résolution

A l'ouverture du fichier joint au challenge (`lost_curve.py`), on observe le déroulement du challenge :

1. Le programme tire des nombres entiers aléatoires `p` , `a` , `b` , `xP` puis nous fournit 2 points `P` et `Q` sur la courbe d'équation `y**2 = x**3 + a*x + b % p` .
2. Nous devons déteminer `a` , `b` , `p` et les retourner au serveur.

Le point `Q` n'est pas tiré au hasard, en effet il est généré par la relation `2*P = Q` .

N'étant pas particulièrement spécialiste des courbes elliptiques, j'ai récupéré sur Wikipédia les formules d'opérations sur les points :

- Pour calculer `R = 2*P` , on pose `s = (3*xP**2 + a)/(2*yP)` , et ainsi :
  - `xR = s**2 - 2*xP % p`
  - `yR = -yP + s*(xP - xR) % p`
- Pour calculer `R = P + Q` , on pose `s = (yP + yQ) / (xP - xQ)` et ainsi :
  - `xR = s**2 - xP - xQ % p`
  - `yR = -yP + s*(xP - xR) % p`
- Pour calculer `R = -P` :
  - `xR = xP`
  - `yR = -yP`

Fort de ce savoir, je commence à construire les équations qui découlent de la relation `Q = 2*P` afin de déterminer des relations qui pourraient m'aider à trouver un ou plusieurs paramètres :

- `yQ + yP = s*(xP - xQ) % p`
  - Cette formule nous fournit `s` si on connait `p` . En effet, `s = (yQ + yP) * modinv(xP - xQ, p) % p`
- `xQ + 2xP = s**2 % p`
- `a*xP + b = yP**2 - xP**3 % p`
  - Grâce à cette équation, si je connais `p` et `a` , je connais `b` simplement : `b = yP**2 - xP**3 - a*xP % p` (en effet, `0 < b < p`)

En remarquant `2*P = Q <=> Q - P = P <=> -P + Q = P` , j'ai pu ajouter l'équation suivante qui a étonnamment débloqué ma situation *(je ne pensais pas gagner de l'information en effectuant cette transformation)* :

- `(yQ + yP)**2 / (xQ - xP)**2 - xP - xQ = xP % p`
  Qui est équivalente à :
- `(2*xP + xQ) * (xQ - xP)**2 + (yQ + yP)**2 = 0 % p`

C'est à dire qu'il existe `k` un entier tel que `(2*xP + xQ) * (xQ - xP)**2 + (yQ + yP)**2 = k*p`

Ainsi, en calculant `A = (2*xP + xQ) * (xQ - xP)**2 + (yQ + yP)**2` (toutes les variables présentes sont données dans le challenge), puis en factorisant `A` en facteurs premiers, `p` apparaît (il est reconnaissable par sa taille >= 80 bits).

Par la relation précédemment évoquée `s = (yQ + yP) * modinv(xP - xQ, p) % p`, j'obtiens ainsi `s`. Il en découle `a = s*2*yP - 3*xP**2 % p` par définition de `s`

Enfin, `b = yP**2 - xP**3 - a*xP % p`.

Je dispose maintenant de tous les paramètres, je peux les envoyer au challenge et récupérer le flag.