```
import hashlib
import sys
from Crypto.Cipher import AES

def HenselLift(P, p, prec):
    E = P.curve()
    Eq = E.change_ring(QQ)
    Ep = Eq.change_ring(Qp(p,prec))
    x_P,y_P = P.xy()
    x_lift = ZZ(x_P)
    y_lift = ZZ(y_P)
    x, y, a1, a2, a3, a4, a6 = var('x,y,a1,a2,a3,a4,a6')
    f(a1,a2,a3,a4,a6,x,y) = y^2 + a1*x*y + a3*y -x^3 -a2*x^2 -a4*x -a6
    g(y) = f(ZZ(Eq.a1()),ZZ(Eq.a2()),ZZ(Eq.a3()),ZZ(Eq.a4()),ZZ(Eq.a6()),ZZ(x_P),y)
    gDiff = g.diff()
    for i in range(1,prec):
        uInv = ZZ(gDiff(y=y_lift))
        u = uInv.inverse_mod(p^i)
        y_lift= y_lift -u*g(y_lift)
        y_lift = ZZ(Mod(y_lift,p^(i+1)))
    y_lift = y_lift+O(p^prec)
    return Ep([x_lift,y_lift])



def SmartAttack(P,Q,p,prec):
    E = P.curve()
    Eqq = E.change_ring(QQ)
    Eqp = Eqq.change_ring(Qp(p,prec))

    P_Qp = HenselLift(P,p,prec)
    Q_Qp = HenselLift(Q,p,prec)

    p_times_P = p*P_Qp
    p_times_Q=p*Q_Qp
    x_P,y_P = p_times_P.xy()
    x_Q,y_Q = p_times_Q.xy()

    phi_P = -(x_P/y_P)
    phi_Q = -(x_Q/y_Q)
    k = phi_Q/phi_P
    k = Mod(k,p)
    return k


p =
19930885203825540108513916323464619607890520613882110268999315402502483992713624056070847778485065788387695567162120805830837964237496163664777233360835
853
a =
16964108867870875431022511847406074513505438269132199629926447105117327702973321429130078673611996313382775579316996202239740533714013559789369433905598
853
b =
25572699039046829700992365380317352184627808448171753176973856593770071867850509696769380590881559223280110750816063545696219588626128508166946470023808
20

E = EllipticCurve(GF(p), [a, b])
print(f"checking if the curve is anomalous...")
if E.cardinality() == p:
    print(f"OK\n")
else:
    print(f"The curve is not anomalous exiting with 1...")
    sys.exit(1)

print(f"Calculating dlog...")
G =
E(116123160388014799624007236984278833101301631762913387599112810341705566097434577730323759656204299316754103892205123428346630891266594455237015339223
13238,
144924705599766099096149201503025338906622815398745426398082139829563167108376671146814963363619067187480538845134683432360555453375220557131410766159672
43)
P =
E(716447455218443211337138354910360820310039222467099750548875601978669948877653369670984450493776038314939273990792875030477209738905138162016103972822
887,
257763683540484460608659351365624287240962911508468079700967729803003501905607466042840886513230706344244098693799875211562972886027280031137684107683029
7)
dlog = int(SmartAttack(G,P,p,4096))
print(f"SUCCESS, dlog = {dlog}\n")

Q = dlog*P
secret = str(Q[0])
print(f"Secret: {secret}")

hasher = hashlib.sha1()
hasher.update(secret.encode("ascii"))
key = hasher.digest()[:16]
print(f"Key: {key}")

decryptor = AES.new(key, AES.MODE_ECB)
with open("flag.enc", "rb") as f:
    encflag = f.read()
flag = decryptor.decrypt(encflag)
with open("flag.mp3", "wb") as f:
    f.write(flag)
print(f"Flag written to 'flag.mp3'")
```