# Single

首先需要先解出橢圓曲線的係數 a, b，可以直接拿任意兩個點代入方程式即可解出：

```
F.<a, b> = PolynomialRing(Zmod(p))
I = ideal([G.x ** 3 + a * G.x + b - G.y ** 2, A.x ** 3 + a * A.x + b - A.y ** 2])
I.variety()

verbose 0 (3827: multi_polynomial_ideal.py, groebner_basis) Warning: falling back to very slow toy implementation.
verbose 0 (1081: multi_polynomial_ideal.py, dimension) Warning: falling back to very slow toy implementation.
verbose 0 (2264: multi_polynomial_ideal.py, variety) Warning: falling back to very slow toy implementation.
[{a: 9605275265879631008726467740646537125692167794341640822702313056611938432994, b: 7839838607707494463758049830515369383778931948114955676985180993569200375480}]
```

有了 a, b 後確認一下 $4*a^3+27*b^2=0\ (mod\ p)$ 確定這個曲線是 singular curve：

```
a = 9605275265879631008726467740646537125692167794341640822702313056611938432994
b = 7839838607707494463758049830515369383778931948114955676985180993569200375480
(4*a^3 + 27*b^2)%p

0
```

接著就可以解出兩個根 alpha, beta：

```
F.<x> = PolynomialRing(Zmod(p))
r = (x^3+a*x+b).roots()
r

[(7925182757193285961316626419940151258451119718064102936455321651294650340555,
  1),
 (853242911173207820721903052331400912971957115055181874915218687301323932414,
  2)]
```

定義上課提到的 phi function，並驗證是否有 homomorphism：

```
def phi(P):
    t = (alpha - beta).sqrt() * (P.x - alpha)
    return (P.y + t) / (P.y - t)

print(phi(point_addition(A,B)) == phi(A) * phi(B))

True
```

透過 `discrete_log` 解出 dA 後，然後就可以拿到 key，接著再對密文做解密即可取得 flag：

```
dA = discrete_log(phi(A), phi(G))
dA
```

```
1532487521612462894579517163606359285989568203515734083099567402780433190052
```

```
k = point_multiply(B, dA).x
k
```

```
19524483693942273038463516851242397572200553783446891974284143156190051301 2
```

```
import hashlib

k = hashlib.sha512(str(k).encode('ascii')).digest()
flag = bytes.fromhex('1536c5b019bd24ddf9fc50de28828f727190ff121b709a6c63c4f823ec31780ad30d219f07a8c419c7afcdce900b6e89b37b18b6daede22e5445eb98f3ca2e40')
bytes(ci ^^ ki for ci, ki in zip(flag, k))
```

```
b'FLAG{adbffefdb46a99fad0042dd3c10fdc414fadd25c}\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
```