# Q1 Crypto

This CTF challenge is related to the Discrete Logarithm Problem (DLP), focusing on how its security is established. The challenge tests your understanding of asymmetric encryption by using an outdated encryption bit-width. (How to generate this?)

```
// Crypto challange
// Assume only you know the password
// Prove code below is not safe
if (info.try) {
    try {
        let a = BigInt('0x' + info.try)
        let b = BigInt("0xb6d733a404d0b06e51dcf52fec53b6b9ed807b3bdc13dbe33e5e59182f66b733")
        let c = BigInt("0x3e9")
        let d = "4384742de6012452302030a8c48605374070da2f41d5847b066bcd94f32a05e0"
        let result = ((a ** c) % b).toString(16)
        if (result == d) {
            let hex = Buffer.from(info.try, 'hex')
            socket.send(JSON.stringify({ flag: hex.toString('utf8') }))
        }
        else {
            socket.send(JSON.stringify({ flag: "Zzzzzzzz...." }))
        }
```

*Figure 1 DLP*

## POC:

1. **Factorizing large numbers**
   Tools: yafu (or any tool)
   Key in
   factor(0xb6d733a404d0b06e51dcf52fec53b6b9ed807b3bdc13dbe33e5e59182f66b733)
   We get

```
***factors found***

P39 = 244797265212401102686995522653336482037
P39 = 337835338562002625014208649165305613959

ans = 1
```

*Figure 2 p and q*

   Prime1 = 244797265212401102686995522653336482037
   Prime2 = 337835338562002625014208649165305613959

2. **Calc φp = (**Prime1**-1)*(** Prime2**-1)**
   φp =
   827011669720838739635026813210919042672528518814801496267512137241208178584 88

3. **Start calc G (true message)**

Tools: [RDLP (windows only)]

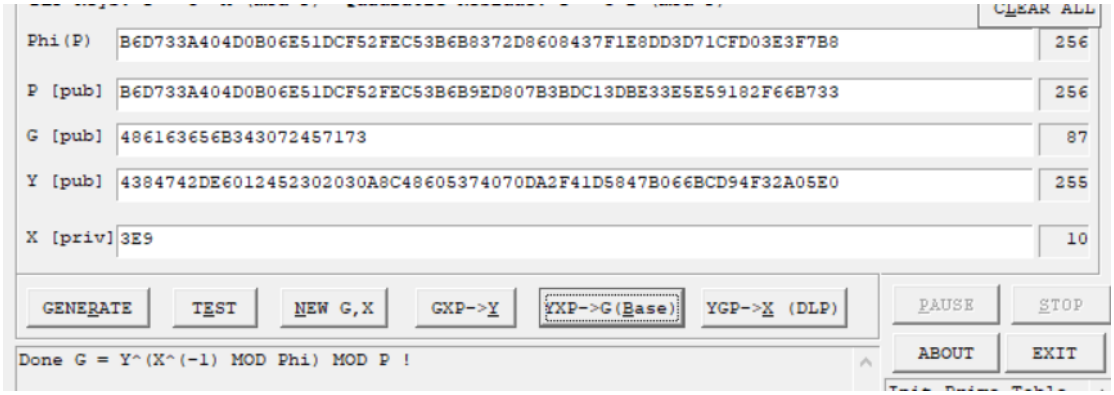


*Figure 3 get G*

We get answer 486163656B343072457173.

## 4. Verification


*Figure 4 final flag*

[Attack Script]