# 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 The challenge tests your understanding of asymmetric encryption by using an outdated encryption bit-width.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?)](https://github.com/FrogGuaGua/SQRPRGRM/blob/main/CTFanswer%26design/Q1/design.js)  


Figure 1 DLP

## POC:

1. **Factorizing large numbers**

Tools: [yafu](https://github.com/bbuhrow/yafu) (or any tool)

Key in

factor(0xb6d733a404d0b06e51dcf52fec53b6b9ed807b3bdc13dbe33e5e59182f66b733)

We get

A screenshot of a computer screen

Description automatically generated

Figure 2 p and q

Prime1 = 0d244797265212401102686995522653336482037

Prime2 = 0d337835338562002625014208649165305613959

1. **Calculate φp (Euler's Totient Function)**

We use Euler's Totient Function to determine the uniqueness of G because a ([figure 1](#Figrue_1)) is not a prime.  
φp = (Prime1-1) \* ( Prime2-1)

= 0d82701166972083873963502681321091904267252851881480149626751213724120817858488

= 0xB6D733A404D0B06E51DCF52FEC53B6B8372D8608437F1E8DD3D71CFD03E3F7B8

1. **Start calc G (**a in [figure 1](#Figrue_1)**)**  
   Tools: [RDLP (windows only)](https://github.com/FrogGuaGua/SQRPRGRM/tree/main/CTFanswer%26design/Q1/RDLP_v1.07)

A screenshot of a computer

Description automatically generated

Figure 3 Calc G

Put them into the RDLP  
Phi(p) <- φp in part2

P <- b in [figure 1](#Figure1)

Y <- d in [figure 1](#Figrue_1)

X <- c in [figure 1](#Figrue_1)  
 Click the YXP-> G

We get answer G (a in [figure 1](#Figrue_1)) = 0x486163656B343072457173.

1. **Verification**

Write an [Attack Script](https://github.com/FrogGuaGua/SQRPRGRM/blob/main/CTFanswer%26design/Q1/answer.js)

A screenshot of a computer program

Description automatically generated

Figure 4 Attack script

Run it, and now we get the correct flag.



Figure 5 final flag