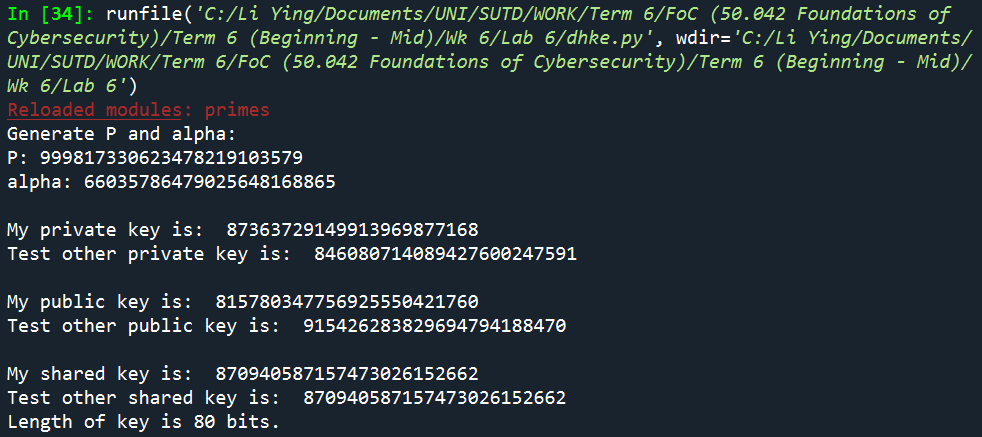
**50.042 FCS Lab 6 Section 4 Write-up**

**Demo a key exchange using DHKE protocol**



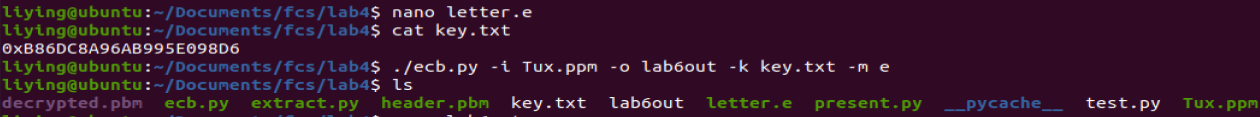
Generated shared key = 870940587157473026152662

**Use PRESENT with ECB mode from the previous lab to encrypt a message using the shared keys. Note that PRESENT requires the key to be either 80 or 128 bits.**

Using the shared key of 80 bits from the previous part, generated shared key = 870940587157473026152662

Converting shared key to base 16 because input key file has to contain key in hex format. Key in base 16 = 0xB86DC8A96AB995E098D6

Using ecb.py (and present.py) from Lab 4,





The decrypted file gives the same original image when displayed.

**Advantage and Disadvantage of DHKE**

Advantage

* Strong against passive eavesdroppers
* Diffie-Hellman protocol is secure because of Discrete Logarithm Problem
* Attacker cannot find a from (ga mod p) although she has g and p
* Computation of Discrete Logarithm Program is hard because modular multiplication is expensive. Algorithms for DLP solving exist but they are not efficient enough for sufficiently large values of p

Disadvantage

* Weak against Man in the Middle attacks
* If the attacker can see packets and change the packets or swap the order of the packets, he can carry out MitM attack, so DHKE is not safe in this case.
* Eve will retrieve the partial key from Alice and Eve will use the same protocol and generate a different key (instead of b) to sync with Alice. Alice will think this key she receives is from Bob. Alice and Eve will have a shared key and communicate using this, and Alice will think she is communicating with Bob.
* The same scenario will play out between Eve and Bob. Eve and Bob will establish another pair of keys to get a shared key.
* Eve can forge messages and neither Alice nor Bob will know that their messages are being intercepted by a third party.
* ALSO, if the prime and the generator are not chosen properly, the generator only generates to a small subgroup, and there will be a vulnerability