# Use the PRF in some secure mode of operation to obtain a CPA-secure encryption scheme

To obtain a CPA-secure encryption scheme using PRF, Let F be a PRF. First, one needs to define a private-key encryption scheme for messages of length n.
The encryption scheme is a collection of three algorithms, namely, key generation algorithm (Gen), encryption algorithm (Enc) and decryption algorithm (Dec). The Gen gives key to both the sender and receiver, the Enc with key, plain text and local randomness gives the ciphertext as the output, and Dec takes the ciphertext and key to output the message.
For the CPA-secure encryption :

1. Gen : on input $1^n$, choose k <- $\{0,1\}^n$ uniformly at random and output it as the key
2. Enc : on input a key where k is belonging to $\{0,1\}^n$ and a message 'm' belonging to $\{0,1\}^n$, choose r <- $\{0,1\}^n$ uniformly at random and output the ciphertext
   $c := \langle r, F_k(r) \oplus m \rangle$
3. Dec : on input a key 'k' belonging to $\{0,1\}^n$ and a ciphertext $c = \langle r,s \rangle$, output the plaintext message
   $m := F_k(r) \oplus s$

The encrypt function first gets the binary representation of the message and then encrypts followed by converting it back to the ASCII representation. The result is encoded as a base64 string. Given a key and a message, it outputs r and the ciphertext.

The decrypt function first gets the binary representation of the ciphertext then decrypts followed by converting it back to ASCII representation. Given the key and the ciphertext, the function deciphers and returns the message.

Following this, the code handles the arguments for generate and verify mode.

The code has been explained in the comments present in the script wherever necessary.