Python3 has many useful libraries (PyCryptodome) that can deal with Cryptography scheme.

Python3 differentiates string and bytes with the `b''` syntax

```
 s1 = b'abcd' # Bytes of 'abcd'
s2 = 'abcd' # String of 'abcd'
```

Here are some of the useful commands that is supported natively in Python3:

- `s.encode()` - Convert string `s` into bytes
- `b.decode()` - Convert bytes `b` into string
- `b.hex()` - Convert bytes `b` to hex string
- `bytes.fromhex("01abcd")` - Convert the hex string `01abcd` to bytes
- `bytes([1,2,3])` - Convert integer list to bytes
- `list(b) - Convert bytes` - Convert `b` into a list of integers
- `i = 0x1235` - Set the value of `i` to be the value `0x1235`
- `i = int("1234ab", 16)` - Convert hex string `"1234ab"` to integer
- `pow(c, e, m)` - Calculate c^e mod m

For most of Cryptography library in python3, they require the plaintext to be bytes and not a string. This is because a string in python3 might have different encoding, but the encoding for bytes is universally UTF-8

Here are some of the useful commands that is supported in PyCryptodome:

- `Crypto.Util.number.long_to_bytes(m)` - Convert integer `m` to bytes
- `Crypto.Util.number.bytes_to_long(b)` - Convert bytes `b` to integer
- `Crypto.Util.Padding.pad(b, x)` - Pad bytes `b` so that the length is multiple of `x`
- `Crypto.Cipher.AES.new(key, AES.MODE_ECB)` - A new AES instances in ECB mode

To dynamically with interact with TCP server, you can use pwntools

```
 from pwn import * # Import pwntools

r = remote("123.123.123.123", 15000) # Connect to 123.123.123.123 at port 15000

s = b'abcde'
r.sendline(s) # Send bytes s to the server
r.sendafter(b'message:', s) # Send bytes s after received bytes 'message:'

r.recvline() # Receive a line from the server
r.recvuntil(b'Nonce: ') # Receive until the bytes 'Nonce: ' from the server
r.recvall() # Receive all bytes until EOF

r.interative() # Change to interative mode
```

Note that all the received message are in bytes. So you might to some conversion if necessary.

You can also change to debug mode with

`r = remote("123.123.123.123", 15000, level='debug')`