Encryption in Python Project

Introduction: As the task was to send a message using the secret key and later also decrypt the message to see whether the message sent was correctly received to the end by using the key, I considered writing a message first which was “I want to be a Quantum Physicist” for example.

After writing the message then I used the QKD function to write the key which I wrote as,

```
key = QKD(len(unencrypted_string)*8)

# Assuming the key is already defined as a list

key_length = len(unencrypted_string) * 8  # length of unencrypted_string * 8 bits per character



# Repeat key to match the length of unencrypted_string * 8

key = key * (key_length // len(key)) + key[:key_length % len(key)]



# Now, the key is adjusted to match the length required

print(len(unencrypted_string),len(key))

print(key)
```

Then after checking that the key was written, I wrote the two important functions for the program which were first to encrypt the message and the other was to decrypt the message,

```
def encrypt_message(message, key):

  # Convert message to binary

  binary_message = ''.join(bin(ord(char))[2:].zfill(8) for char in unencrypted_string)

  print(len(binary_message), len(key))

  #XOR the message with the key

  encrypted_message = ''.join(str(int(binary_message[i]) ^ key[i]) for i in range(len(binary_message)))



  return encrypted_message
```

And to decrypt the message I write,

```
def decrypt_message(encrypted_message, key):

  # XOR the encrypted message with the key

  decrypted_message = ''.join(str(int(encrypted_message[i]) ^ key[i]) for i in
range(len(encrypted_message)))



  # Convert binary back to characters

  decrypted_string = ''.join(chr(int(decrypted_message[i:i+8], 2)) for i in range(0,
len(decrypted_message), 8))



  return decrypted_string
```

And now to check whether both functions work well I then use another snippet of code to check all the messages together, encrypted, decrypted and the original message that the user sent onto the system.

And after running this final cell I get the result as follows

```
Original Message: I want a life.

Encrypted Message:
10000101111101000101111011111110110100011001101101011100110101101111101
00010001011111010110101011001001111011011 1

Decrypted Message: I want a life.
```

In my program, I've converted the message to the binary digits to encrypt the message, there are various other encrypting methods that we can use to encode any message.

Code Snippets: I've not added all the code here, you can check out the whole code notebook on Google Collab here.

Analysis: If there is any eavesdropping then the previous QKD code will help us know that there exists someone in between the connection that has measured some qubits and therefore only even after the key Bob is unable to get the same output as Alice.

Conclusion: In conclusion, the development of quantum encryption is essential for secure communication as it offers unbreakable encryption methods that are resistant to current and future computational attacks. It ensures the security and integrity of communication channels, providing unprecedented levels of security and privacy.

Submitted by
Shreya Satsangi
21.04.24