readyXORnot

Written by ysf

The first cryptography challenge of this years TAMUCTF event reads:

```
original data: "El Psy Congroo"
encrypted data: "IFhiPhZNYi0KWiUcCls="
encrypted flag: "I3gDKVh1Lh4EVyMDBFo="
The flag is not in the traditional gigem{flag} format.
```

If the XOR-encryption is unknown to you, feel free to **read the wikipedia page about xor-ciphers**.

That said, to get the key used for both messages, we need to XOR the original data with the encrypted data. Then we'll use it to decrypt the encrypted flag.

A little caveat might be that the encrypted data (and flag) are encoded in base64. This is spoiled by the = at the end. This makes sense, the XORed data surely contains imprintable chars. So, let's decode them first.

In python it might look like this:

```
import base64
from itertools import izip, cycle

original = "El Psy Congroo"
enc_data = base64.decodestring("IFhiPhZNYi0KWiUcCls=")
enc_flag = base64.decodestring("I3gDKVh1Lh4EVyMDBFo=")

def xor(data, key):
        return ''.join(chr(ord(x) ^ ord(y)) for (x,y) in izip(data, cycle(key)))

key = xor(enc_data, original)
print(xor(enc_flag, key))
```

Which will output FLAG=Alpacaman after start.