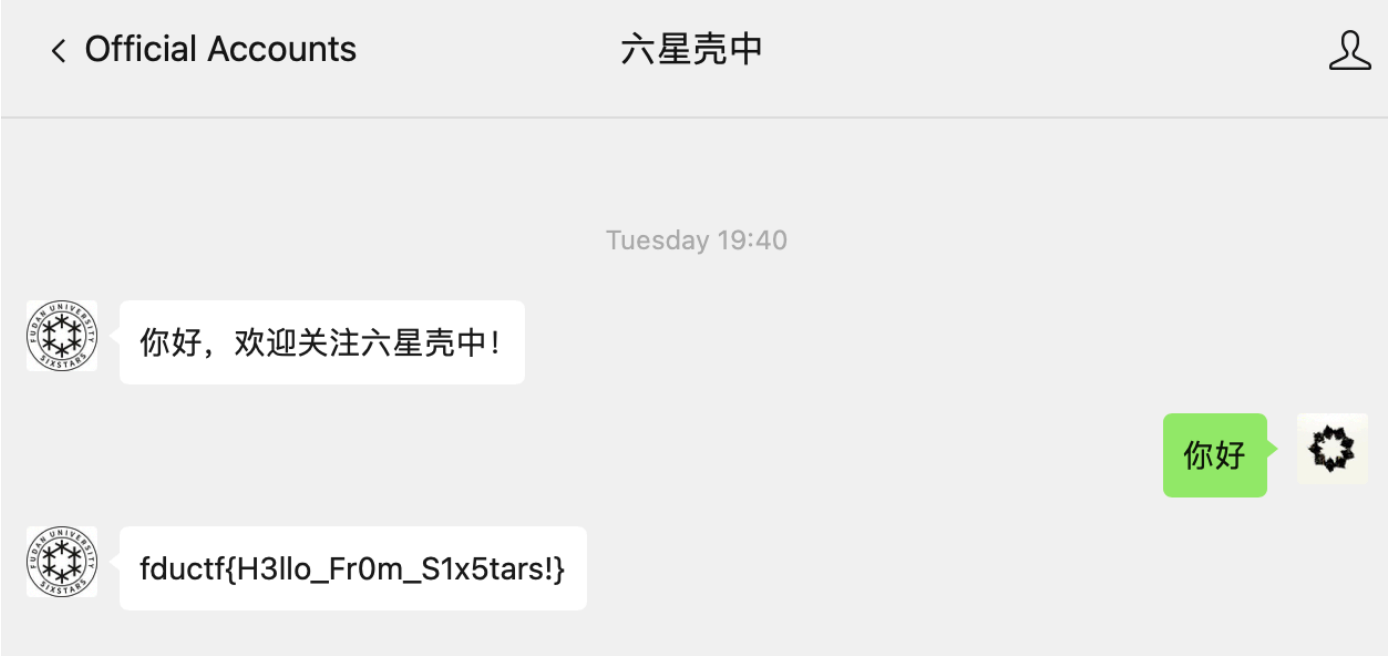
FDUCTF 2024 Writeup

By 8u-nekoyellow

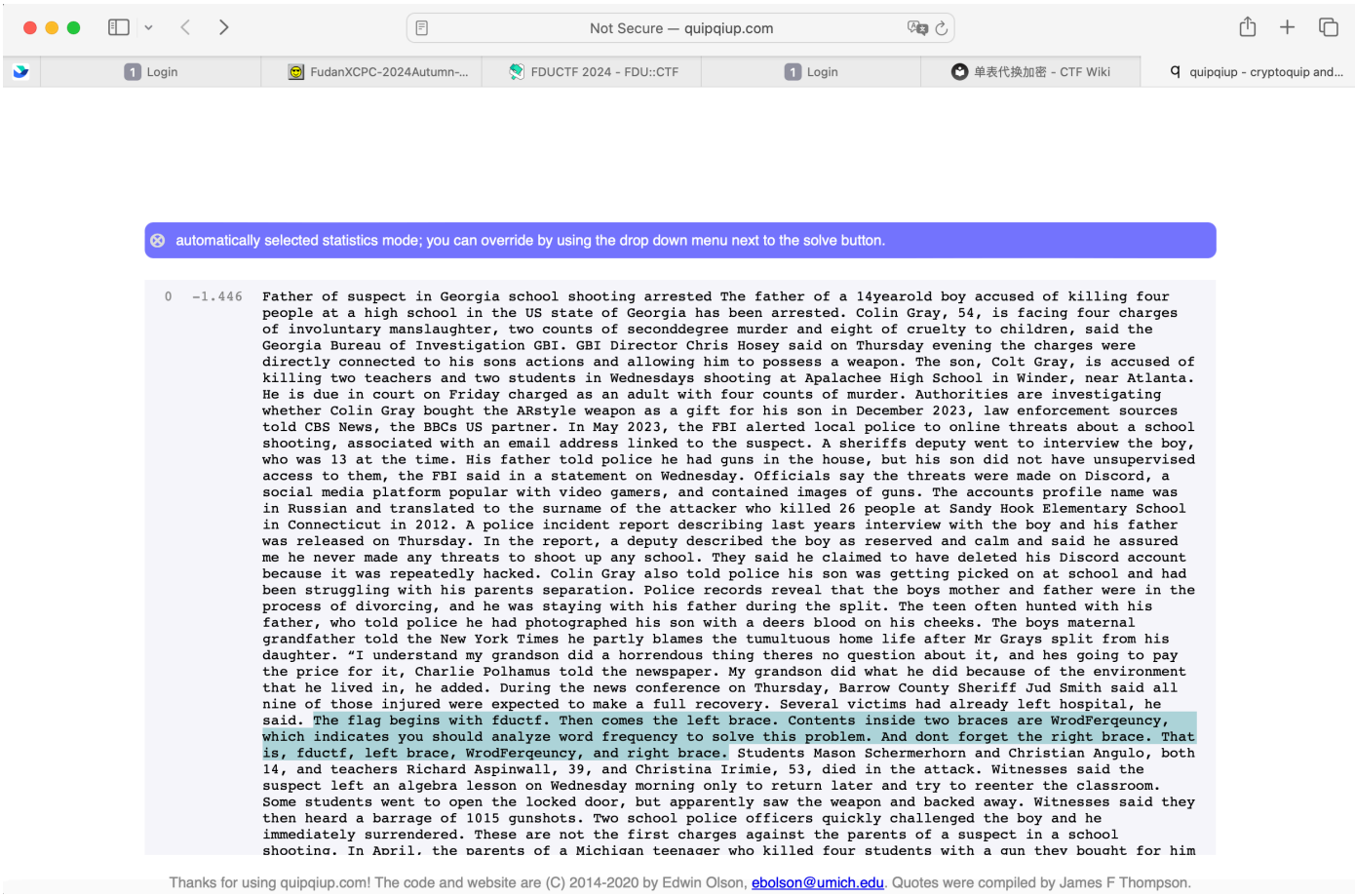
签到

签到。



Alice与Bob的小纸条

使用词频分析（http://quipqiup.com）。



Jeff Dean

进行RSA解密。已知public key e, n，需要推出private exponent $d = e^{-1} \mod \phi(n)$，即e在模$\phi(n)$意义下的逆元。
$\phi(n)$是欧拉函数，表示的是小于等于 n 和 n 互质的数。如果已知n的质因数分解$n = \Pi_i p_i^{k_i}$，可以用公式$\phi(n) = n \times \Pi_i \frac{p_i - 1}{p_i}$计算得到。

对n分解质因数：

```
from sympy import factorint
print(factorint(n))
```

得到

```
# n = p * q
p = 8193423899118349
q =
918549832129642631217357209306156789342787361268206093941796483383789484542708798597295010
966934663904150064134172470988102552209051513911732043909262052289347076966590217809538667
5605831270153783753 59
```

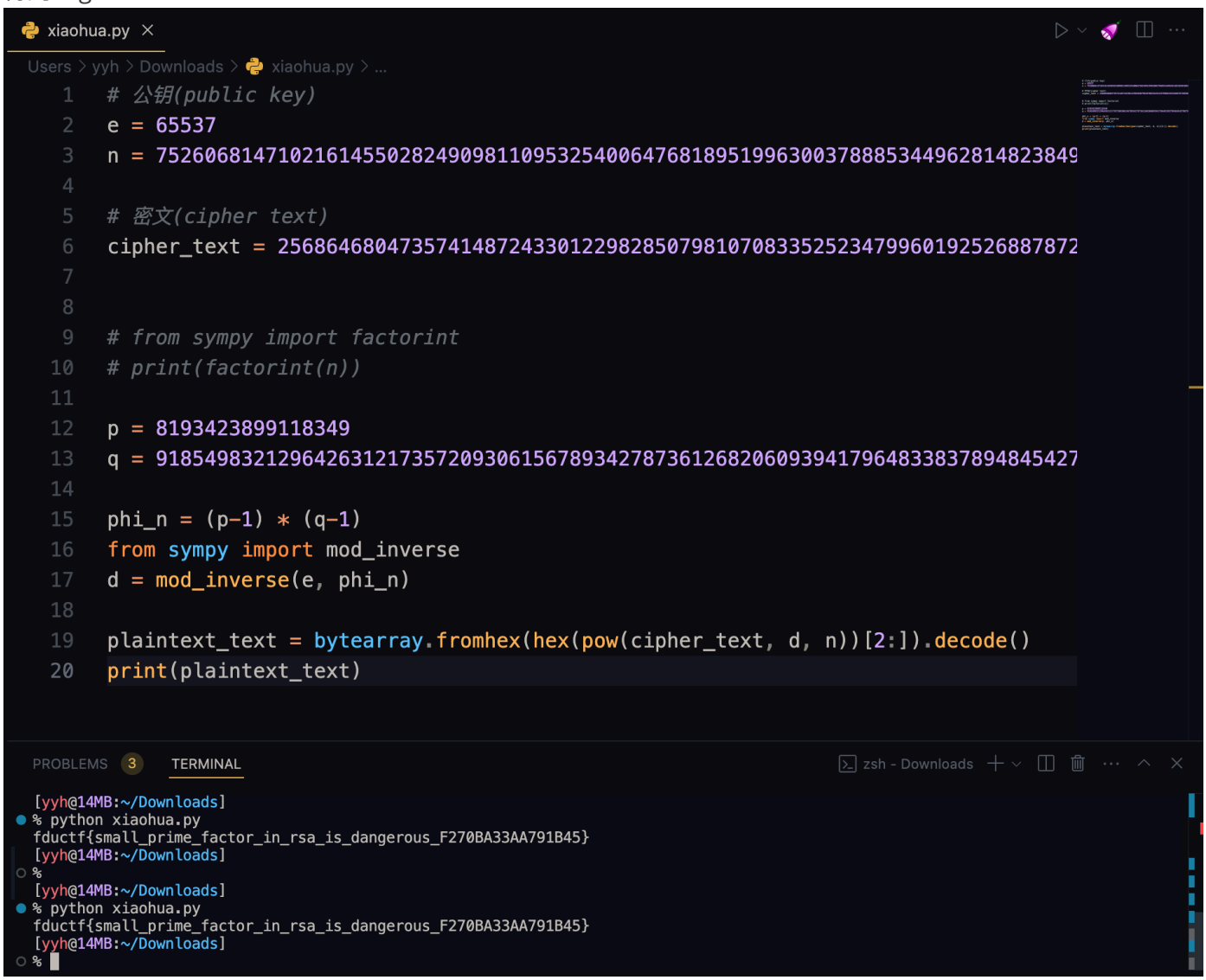
因此

```
phi_n = (p-1) * (q-1)
from sympy import mod_inverse
d = mod_inverse(e, phi_n)
```

最后用$c^d \mod n$解密

```
plaintext_text = bytearray.fromhex(hex(pow(cipher_text, d, n))[2:]).decode()
print(plaintext_text)
```

得到flag：



草率的毕业设计

任务是从源码破解用户名和密码。分析`main.py`可以直接知道用户名为admin（`base64.b64decode("YWRtaW4=").decode('utf-8')`），主要是找密码。分析知道密码会被逐位concat一个4位的salt哈希后与secret数组比较。使用的哈希算法SHA512目前无法破解（无逆运算、冲突概率极小），所以只能考虑暴力。

由于salt长度很短，枚举空间不大（$\sim 100^4$），首先用如下代码穷举找salt：

```
from functools import reduce
```

```python
from itertools import product

for s in product(string.digits + string.ascii_letters, repeat=4):
    s = reduce(lambda x, y: x + y, s)
    ok = True
    for i in range(2): # 看前两位（之前代码写成第一位匹配就过了，不过也得到了正确答案）
        flag = False
        for c in string.ascii_letters + string.digits + "_{}":
            flag |= (sha512(c + s) == secret[i])
        ok &= flag
    if ok:
        salt = s
        break
```

得到`salt = 95qW`。

接下来只要逐位找匹配的字符即可找到密码：

```python
for i in range(n):
    flag = False
    for c in string.ascii_letters + string.digits + "_{}":
        if sha512(c + salt) == secret[i]:
            ans += c
            flag = True
            break
    assert flag
```

得到flag：