Problem 2 - Protocol Failure Exploit

April 12, 2020

[28]:
```
import sys
sys.path.append('../')
import crypto_utils as utils
%load_ext autoreload
%autoreload 2
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

[29]:
```
# Part a explanation of how Oscar can easily decrypt these messages:
print("If Alice uses only the numbers 0-25 and maps the letters directly␣
 ↪corresponding to how the English alphabet is ordered, then the message can␣
 ↪be easily decrypted. You can simply encrypt all the numbers 0-25 and that's␣
 ↪the complete alphabet of possible ciphertexts. Then just work backwards from␣
 ↪the ciphertexts. For example the first ciphertext number is 365, which␣
 ↪corresponds to 21, which is letter V.")
```

```
If Alice uses only the numbers 0-25 and maps the letters directly corresponding
to how the English alphabet is ordered, then the message can be easily
decrypted. You can simply encrypt all the numbers 0-25 and that's the complete
alphabet of possible ciphertexts. Then just work backwards from the ciphertexts.
For example the first ciphertext number is 365, which corresponds to 21, which
is letter V.
```

[30]:
```
# I became in, potentially messed up order problem 1
n = 18721
b = 25
# vanilla, problem 2
alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

for i in range(26):
    print(alphabet[i] + '=' + str(utils.rsa_encrypt(i, b, n)))
```

```
A=0
B=1
C=6400
D=18718
E=17173
```

```
F=1759
G=18242
H=12359
I=14930
J=9
K=6279
L=2608
M=4644
N=4845
O=1375
P=13444
Q=16
R=13663
S=1437
T=2940
U=10334
V=365
W=10789
X=8945
Y=11373
Z=5116
```

[31]:
```
# 365, 0, 4845, 14930, 2608, 2608, 0
# V, A, N, I, L, L, A
```