being-eve.pdf
Owen Barnett and Nick Pandelakis
13 April 2021

A. Diffie Hellman

1. Alice and Bob's shared secret is 12.
2. To solve for Alice and Bob's secret, we observe that 17 and 61 are coprime, so 17 is a generator of the multiplicative group of integers modulo 61. Thus 17^x will generate each integer between 1 and 60 precisely once for 0 <= x <= 60. In python, we input

```
for i in range(61):
    print(str(i) + '. ' + str(((17 ** i) % 61)))
```

3. Using the output of this list, we saw that ‘14. 46’ was output, so Alice’s secret must have been a = 14. Using Bob’s B = 5, we calculated in python

```
(5 ** 14) % 61 = 12
```

   So Alice and Bob’s secret is 12.

4. This method would break because the for loop runs in exponential time for the number of digits in the given prime. If, for example, the given prime were 20 digits long, we would have to run our for loop 10^20 times.

B. RSA

1. Dear Bob, Check this out. https://www.schneier.com/blog/archives/2017/12/e-mail_tracking_1.html Yikes! Your friend, Alice

2. Step 1: We factor 4661 to obtain p = 59 and q = 79
3. Calculate (p - 1)(q - 1) = 4524
4. We want to find d such that 31 x d = 1 mod 4524
    a. We ran the following python code to find d

```
count = 1
x = 31
while x != 1:
    x = (x + 31) % 4524
    count += 1

print(count)
```

5. This returned d = 2335

6. We then ran the following python code to obtain the message

```
def decrypt(encrypted):
return (encrypted ** 2335) % 4661

cypher = [2677, 4254, 1152, 4645, 4227, 1583, 2252, 426, 3492, 4227, 3889,
1789, 4254, 1704, 1301, 4227, 1420, 1789, 1821, 1466, 4227, 2252, 3303, 1420,
2234, 4227, 4227, 1789, 1420, 1420, 4402, 1466, 4070, 3278, 3278, 414, 414,
414, 2234, 1466, 1704, 1789, 2955, 4254, 1821, 4254, 4645, 2234, 1704, 2252,
3282, 3278, 426, 2991, 2252, 1604, 3278, 1152, 4645, 1704, 1789, 1821, 4484,
4254, 1466, 3278, 1512, 3602, 1221, 1872, 3278, 1221, 1512, 3278, 4254, 1435,
3282, 1152, 1821, 2991, 1945, 1420, 4645, 1152, 1704, 1301, 1821, 2955, 1604,
1945, 1221, 2234, 1789, 1420, 3282, 2991, 4227, 4410, 1821, 1301, 4254, 1466,
3454, 4227, 4410, 2252, 3303, 4645, 4227, 3815, 4645, 1821, 4254, 2955, 2566,
3492, 4227, 3563, 2991, 1821, 1704, 4254]

plain = []

for i in range(len(cypher)):
        plain.append(chr(decrypt(cypher[i])))

print("".join(plain))
```

   Which output the correct message.

7. This method would fail in two places. The first is that, for sufficiently large n, factoring n into two primes p and q would be slow. It would then also likely fail in step 4 when we tried to find d. Both of these grow exponentially as the number of digits increases.