Maksim Semchuk
CS 491
Homework #2

1. SHA-3 with a untruncated bitlength of 256 or 512.
2. The number of integers between 1 and n which are coprime to n.
3. It is harder to factorize large prime numbers, then it is small numbers.
4. Given a key with 256 bits:
   a. 2^{256} potential keys – 2^{255} before success is expected.
   b. Assuming: [(2^{256} keys ×16) / (4.4201×10^17 Flops/s)] ≈ 1.329107×10^{53} years of compute time for the Japanese Fugaku Supercomputer. I would say, yes it is a good sized key since the age of the universe is estimated to be only 9.6×10^{42} years. Even assuming that they manage to guess it in half the time, it will still be longer then the current age of the universe. It will take so long to decrypt that, by the time you get it, it wont matter anymore… and didn't for the past several trillion years.

Part 2:

1. Question 1: What exact command did you use in Step 3?
   - I used:
     - openssl enc -aes-128-ecb -in testfile.txt -out testfile.txt.aes -K 'E33202510575DF98CD66D5F35A1915D0' -iv '582221FEB84119C54FC41FBED8E9D778'
       1. Though, since chaining in between blocks is NOT used, the IV is not needed and is ignored due to the fact that aes in ecb mode is used.
2. Question 2:
   - semchuk2@snares:~/Desktop/CS491/hw2$ hexdump -C testfile.txt.aes
   - 00000000 0b f9 f9 34 2a 40 f1 7d 4e 91 76 4c e9 af e7 14 |...4*@.}N.vL....|
   - 00000010 f4 99 86 e0 93 da 7e 24 31 f4 42 50 08 54 db 5e |......~$1.BP.T.^|
   - 00000020
   - semchuk2@snares:~/Desktop/CS491/hw2$ hexdump -C testfilecopy.txt.aes
   - 00000000 38 8e 7c 28 c4 40 ed a7 1a 61 6f 6f 0c 1e 5e b7 |8.|(.@...aoo..^.|
   - 00000010 f4 99 86 e0 93 da 7e 24 31 f4 42 50 08 54 db 5e |......~$1.BP.T.^|
   - 00000020
   - the first set of 16 bytes are different, but the next 16 are the same.
3. What differences do you observe in your results? How does this difference impact security?
   - semchuk2@snares:~/Desktop/CS491/hw2$ hexdump -C testfile.txt.aes
   - 00000000 22 86 a9 4b f3 5e 25 3c 0d e3 d2 43 a5 e1 45 43 |"..K.^%<...C..EC|
   - 00000010 de b9 48 43 5c d4 01 8b d8 95 44 03 92 14 c7 5c |..HC\.....D....\|
   - 00000020
   - semchuk2@snares:~/Desktop/CS491/hw2$ hexdump -C testfilecopy.txt.aes
   - 00000000 57 8e 0d 16 05 0d 02 0c 66 ec 02 7f b4 5f 0b 9e |W.......f...._..|
   - 00000010 15 54 e7 ae fc bb 54 e6 af 00 a2 f9 cb fc ba 98 |.T....T.........|
   - 00000020
   - Here the lines are all completely different, and no byte arrays match. This impacts security because even a small change in the file compounds, leading to a completely different

looking encryption hexdump, and not allowing any collision type attacks where the attacker can use similar outputs to reverse engineer the key in order to decrypt the file.