Quiz 5 student name: Kevin Martin

1: 20 what is secret-key Encryption, is it secure enough and why, or why not?

Secret-key encryption is one of the three pillars of encryption which obscures the original message.

It is **not** secure because it easy to decode by counting the frequency of the letters. In this way, someone can begin to determine which letter is represented by the encrypted letter. Modern computers can speed up this process immensely.

2: 25 what is mono and poly alphabetic substitute cipher? What is the famous example for poly? Why is it is hard to decode?

Mono alphabetic substitute cipher, also known as the Caesar Cipher, is a single permutation mapping system where each letter is represented by only one other letter. For example, all “B’s” will be encrypted as “W’s”.

Poly alphabetic substitue ciphers use multiple permuations, so each letter of an encrypted word may follow a different permutation scheme. This makes it much harder to decrypt, thus increasing security.

The famous example of poly substitute is the Enigma machine.

This is more difficult to decrypt because the total number of the tables to decrypt is 26^4 (in English). This equates to 10^{114} possible combinations (approximately 380 bits which is quite secure) if the wiring is secrete, and if the wiring is known, this is reduced to 10^{23} combinations (76 bits, still secure).

3: 15 what is DES? How does it work(briefly)? What is AES? Why is AES better?

DES is the standard Data Encryption Standard, the single standard for encryption software. It is a standardized protocol to encrypt data. It was initially developed by IBM and then revised with the NSA so they could break it. It initially used 64 bits, but the first 8 bits were not actually part of the algorithm, so it was really 56 bits, or 2^{56} possible combinations.

DES works by taking 16 rounds of “scrambling” or permutation functions, which eventually give out a single output. The input is a 32 bit expansion function, combined with a 48 bit subkey, which both get combined through an XOR gate. The result gets introduced through a perumation function, which produces a 32-bit half-block output.

AES is Advanced Encryption Standard and it uses the Rijndael algorthm, developed in 2001.

AES is better than DES because it does not just have one key size, but multiple key sizes: 128 bit, 192 bit, or 256 bits.

4:10 what’s the problem with the simple encryption block by block? Which lab task this week is to show you the reason?

The problem with simple encryption block by block is that it is not secure. Similar to decrypting a cipher text, you can infer information from looking at multiple blocks. The individual blocks are secure, but taken as a whole, they present information about the entire message.

The lab task this week that shows us this reason is Task 3, which asks us to decrypt a bitmap image. By using block by block encryption, the entire image is still able to be observed, although it is now slightly distorted from the original.

5: 30 What are those Encryption modes discussed in the book and lectures? Brief explain the encryption method for each mode ECB, CBC, CFB and OFB and and if you change one bit in second block of encrypted file, when you decode it, how many and which blocks stay the same and how many and which blocks get corrupted.

The modes discussed in the book and the lecture are Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), and Counter Mode (CTR).

ECB is block by block encryption. While each block is safe, the entire message may not be safe as each block can provide clues to the message as a whole. Changing one bit in a block here will not impact the second block, but will corrupt the entire block.

CBC is that the result of each previous cipher encryption is fed into the encryption of the following (through XOR gates). So the results are essentially chained together. The very first block is fed an “initialization vector” (IV), also through an XOR gate and it is public. This is so the hardware does not need to be different for the first input. Changing one bit in the second block will impact the current block and completely corrupt it. Also, it will impact one bit of the **next** block as XOR is bitwise operation and will be corrupted on the related bit. But all the following blocks will not be corrputed at all.

CFB is similar to ECB, except the IV is encrypted, and that key encrypts the plaintext. However, the plaintext is also split up in a stream, and can be encrypted bit by bit. If the stream is truly bit by bit, then corrupting one bit won’t affect any other bits since it is encrypted by itself.

OFB is that the XOR is moved after the encryption of the IV. This allows for parallel encryption, and generate the entire encryption sequence. Then, as the data is ready, the data can be fed where it is needed, encrypted, with offline help. Similar to CFB, OFB will not have any blocks impacted by the corruption of a single bit as the encryption is handled independently of the data.