Arika Khor
Spring_CS-5173-001
Dr Zhao

Assignment 1

1.

- **Availability**: Information can be reached at all times with the right keys. Eg. information queried is always available (no DDOS attacks preventing queries from reaching the server).
- **Integrity**: Assurance that information hasn't been modified, or if it has, an audit is kept with what is changed.
- **Confidentiality**: Protect information access, only being exposed to users who should have access.

1b) Most important: **confidentiality** given that a data breach would expose the health information of many patients (which may result in a data breach lawsuit).

1c) Both **integrity** and **availability** are equally important. Generally one would think about being able to access the alert system at all times, showing the importance of **availability**, however, to prevent others from sending unauthorized messages on the emergency system (such as turning on the tornado sirens when they shouldn't be) it would be more important to focus on **integrity**. (all sources from Lecture 1&2 slides)

2. In explaining Distributed Denial-of-Service (DDoS) attacks, it's important to understand Denial of Service (DoS) attacks. DoS attack is some attempt at "flooding" a host server from one IP/machine to overload the host server rendering the machine or network resource unavailable to the intended users. In a DDoS attack, there is more than one machine/user (usually from a botnet) that floods the host server, having a distributed load (instead of a single machine) attacking a server. DoS attacks were mostly used before firewalls were implemented, then after servers gained protection DDoS attacks were invented to go around these measures. Cite: https://security.stackexchange.com/a/22816/282055 and lecture 1 slide.
3. Yes, it's easily possible to break the Vigenère cypher (assuming the language is in English) in fact, one can break this cypher by hand (granted it would take a very long time but is possible). If the cypher has a lot of A's in it, one could assume they are E's in normal text (since E is the most common letter). Then, because there are a total possible 26^n keys, where $n =$ key length, it's easy to extrapolate the final plain text. Basically, once one knows the key length, they may treat the cypher as a Caesar cypher instead and solve from there. The process would look something like the following: Finding the key length (with Kasiski Examination), dividing the cypher, then

going through with Frequency Analysis on each group and decrypting the cypher from there. Cite: https://crypto.stackexchange.com/a/73064 and Lecture 3 slides

4. In principle (and definition), a perfectly secure system cannot be cracked as it is designed to be "perfect" against any attacks. That means even with a theoretical infinite computational power, it would still not be possible to crack the cypher. Of course, in practice, there is no such thing as a system that cannot be cracked given that there are always multiple points of failure (such as a key being lost, stolen, or reused). As from the Lecture 4 slides:

   – A perfectly secure system
   - does NOT mean there is no chance to get the plaintext!
   - makes sure that no attack strategy is better than a random guess!

5. Proof: (Citation from Lecture 4 Slides and https://crypto.stackexchange.com/a/20749)

$$Pr[C=c] = \sum Pr[C=c|M=m'] \cdot Pr[M=m'] = \sum Pr[K=m' \oplus c] \cdot Pr[M=m'] = \sum 2^{-}$$

- (from bayes theorem)

$$Pr[M=m|C=c] = \frac{Pr[C=c|M=m] \cdot Pr[M=m]}{Pr[C=c]} = \frac{Pr[k=m \oplus c] \cdot Pr[M=m]}{2^{-n}} = \frac{2^{-n} \cdot}{}$$

6. No, Alice's design can still be broken. In replacing the One Time Pad method with a bit-wise AND `&` instead, we would obtain something like the following:

$$Pr[C=c] = \sum_{M} Pr[C=c \mid M=m] \cdot Pr[M=m] = \sum_{Pr[K=m' \wedge c]} \cdot Pr[M=m'] = \sum 2^{-n} \cdot Pr[M =$$

$$Pr[M=m \mid C=c] = \frac{Pr[C=c \mid M=m] \cdot Pr[M=m]}{Pr[C=c]} = \frac{Pr[K=m \wedge c] \cdot Pr[M=m]}{2^{-n}} = \frac{2^{-}}{}$$

This is quite an intuitive problem if one were to think about it. Basically, if a cypher character is 0 and the key is 0, then the plaintext would result in a 0 or 1 (with a 50/50 chance). Say we took the example input string of: 001101. Then taking M ^ K: 00110^101110=001100 reavling the first 2 bits and thus compromising the cipher. This could be explained with a probability formula like so:

```
P( C=1 | M=0 ) = P( K AND M = 1 | M=0 )
       = P( K AND 0 = 1 | M=0)
       = P( 0=1 | M=0 )
       = 0 ≠ 2^-1
```

- Source: https://crypto.stackexchange.com/a/100827

7. No, key reuse in a One Time Pad will result in a loss of perfect security and can be broken. As a reference to using a "many time pad attack", the steps to attack would be something like the following:

```
1) Guess one word appearing in the message
2) encode word from (1) to hex string
3) XOR both cypher texts
4) XOR hex string from (2) at each XOR position of both cyphers from (3)
5) When (4) is readable, guess the English word and expand the crib search.
If not readable, try XOR of crib word at the next position
```

   - Source for attack steps: https://travisdazell.blogspot.com/2012/11/many-time-pad-attack-crib-drag.html

The proof in pointing to the original proof from problem 5, we would have the two cyphers, C_1C_2, where would be some variation of $C_i = M_i \oplus K$, then knowing `i` cyphers (where i > 1), they could compute the XOR of both ciphertexts like so: $M_1 \oplus M_2 = C_1 \oplus C_2$, hence if the attackers knows the structure of the plain text (assuming it's English), then can be able to observe patterns in both keys and result in a few of the values being compromised. This problem is quite similar to no.6 in the sense that perfect security is lost from giving the end user a few key parts of unencrypted data in order to figure out the last bit of data. Thus the reuse of one-time pad keys will result in an insecure system.

$$Pr[C = c] = \sum Pr[C = c|M = m'] \cdot Pr[M = m'] = \sum Pr[K = m' \oplus c] \cdot Pr[M = m'] = \sum 2^{-n} \cdot P$$

More formally, this could be displayed as: $C_1 = M_1 \oplus K$ $C_2 = M_2 \oplus K$, then $C_1 \oplus C_2 = M_1 \oplus M_2$, hence releaving the relationship between both plaintexts.