Haoshu Yuwen
Hy2892

Homework 3

Lecture 34:
1. Messaging speed is bound by the transfer capacity. If you can only transmit so many bits per second physically, then you can only transmit the number of symbols equal to capacity divided by number of bits per symbol (C/h). This assumes there is no noise whatsoever so in reality, the messaging speed is even slower than that.

2. Increasing redundancy on a noisy channel will increase the likelihood of the receiver receiving the correct message. Since redundancy implies that a slightly modified version of the message is still decoded, this removes the detrimental effects of noise corruption.

Lecture 35:
1. –(log.10)

2. Character frequencies change with variables such as context and length.

3. A zero order model assumes that all symbols are equally likely in a random message. In a first order model, probabilities are assigned to each individual symbols that are different. For higher order models, probabilities are assigned to different combinations of individual symbols.

Lecture 36:
1. Computing prior possibilities requires that the calculator has sufficient data over a broad set of examples. Sometimes this is just not available.

2. If the receiver has no knowledge regarding the message, it is impossible to determine whether the message received has any superfluous information.

3. If the information can be conveyed using a coding that matches the entropy and no less, then there is no redundancy.

Lecture 37:
1. For the casual observer with no knowledge of the coding scheme, this is impossible to decipher. Therefore, we can see that cryptography is an effective tool to protect sensitive information.

2. A key significantly reduces the time it takes to decode a message.

3. It makes it unreadable to the casual observer.

4. Redundancy makes it possible to observe patterns in the encrypted message, which eventually leads to guesses at what each encrypted symbol means.

Lecture 38:
1. P

2. E(P, Ke)

3. Recognizing patterns could potentially lead to discovering the encryption algorithm which would make decrypting any future messages that much easier.

4. Languages have particular patterns that may translate even when encrypted. For example, in English, sentences are broken up into words separated by spaces. An encrypted message may still retain the pattern of having a distinct symbol to represent the space.

Lecture 39:
1. For some algorithms, the key range is literally so large, that to brute force them would take more time than the average lifespan.

2. Assuming the key is an n bit string, there are 2^n-1 possible keys. If you brute force all of them, that would be approximately 2^n-1 operations assuming K is small enough of a file.

3. Without substitution, there would essentially be no encryption at all. Without transposition, the substitution pattern would easily be deduced. (Or at least more easily deduced)

4. Confusion is basically substitution and diffusion is basically transposition.

5. They are both equally important.

Lecture 40:
1. In mono-alphabetic encryption, each symbol is replaced with a set other symbol whereas in poly-alphabetic substitution, each symbol may be replaced with varying symbols depending on where the symbol is in the message.

2. A symbol to symbol mapping

3. If there are k letter in the alphabet, then the 1st letter can be mapped to one of k options, then the 2nd to one of k-1 options, and so on resulting in k! possible mappings.

4. They key is a number value indicated the offset.

5. Including the space, 27 possible keys.

6. No.

7. Find the column in the chart corresponding to the key letter then go down till you find the encrypted symbol. The original letter is the label of the row.

Lecture 41:
1. Without knowing if it's a simple substitution, you have to assume that each x and y could be mapped to any of the 26 other letters and therefore there are 26*26*26 possibilities.

2. Because if we know it's a simple substitution, then y must map to the same letter. Also, y cannot map to the same letter as x. Therefore, there are 26 choices for x and 25 choices for y.

3. No because no matter what cypher, there must be a pattern or else decryption if impossible. If there is a pattern, then by either brute force or sheer dumb luck, eventually it will be discovered.

Lecture 42:
1. Because since the key is the same length as the message, every single possible message could map to the encrypted message depending on the varying key.

2. If it's not random, then you can simply figure out the key.

3. If you cannot securely pass a key to the recipient, then outsiders can simply use the key to decrypt your message.

Lecture 43:
1. Without using substitution at the same time, transposition does not replace the original letters. Therefore, an English encryption would still be using English letters, etc.

Lecture 44:
1. The one time pad is a symmetric encryption process because the same key can be used for both encryption and decryption.

2. Key distribution is getting the key to the intended target. Key management is securing a large amount of keys and preventing unauthorized access.

3. No. Encryption is done with the public key and this encryption process is not symmetric. Therefore, the only the attacker can do with Ks is to encrypt the already encrypted information again.

4 It depends on the context.

Lecture 45:

1. It's harder to tamper with the data without detection.

2. If you mess with the encrypted file, you will mess with the decryption in a meaningful way, which seems not so good if you ask me.

3. You can change data without exposing the data.

Lecture 46:
1. subBytes: It replaces a symbol with a predefined replacement.

2. shiftRows: it shifts the rows.

3. The mixColumns step requires multiplying each column with a fixed array.

4. The code is broken up into blocks to process. Each block goes through 4 rounds of processing.

5. The more rounds, the more difficult it is to break.

Lecture 47:
1. If blocks of data are repeated often, then a pattern can easily be deduced.

2. Randomize the blocks as well as what's in the blocks.

3. If you notice a pattern, you can easily find the first block.

3. Key stream generation is more like a one time pad.

Lecture 48:
1. The private key.

2. If it is extremely difficult to go in reverse, then an encryption is very difficult to break.

3. The public key is public, so anyone can use it. Additionally, have the public key doesn't mean the user can decrypt.

4. C (Encrypted File)

5. Symmetric encryption processes are significantly faster.

Lecture 49:
1. Yes. That's just the way it was designed.

2. It is extremely difficult to factor a significantly large number into two primes.

3. Yes by brute force, but it would take a very long time.

4. Without the private key, decoding is practically impossible.

5. Because everyone has access to A's public key so anyone can send him an encrypted message.

6. Because if he can decode it with B's public key, then it must have come from B because B is the only one with access to Kb.

7. Since the public key is public, everyone can decode a message encoded with the private key.

8. Use another key to sign the message.

Lecture 50:
1. Efficiency is important.

2. For weak, it must be hard to find a m2 that hashes the same as m1 if you already know m1. It doesn't necessarily mean it's hard to find just two general messages that hash the same.

3. In preimage resistance, you are trying to match a hashed message with the original. In second preimage resistance, you are trying to match the hashed message of two different messages.

4. There are 1.25*(2^128)^.5 arguments that will result in distinct hash values.

5. There are 1.25(2&160)^.5 arguments that will result in distinct hash values.

6. uh…

7. It's difficult to find two messages that hash to the same value.

8. Hash the message and encrypt the result.

Lecture 51:
1. No because anyone can decrypt the outer encryption with R's public key and again the inner with S's public key.

2. Yes. No one but S could have performed the outer encryption and only R can decrypt the inner encryption.

3. No

4. Confidentiality and authenticity because we need to be sure the sender is the correct sender AND that nobody but the recipient should be able to see what the key is.

Lecture 52:
1. They would not be able to break the key.

2. We can calculate b.

3. We could calculate a.