Name: Yun Wen Chen
EID: dc27863
CS Login: dchen
Email: dianachen@utexas.edu

Lecture 34

1. Because C and H are ideal values. C is the rate of transmission of a noiseless channel and h is the measure of the most efficient possible encoding of a language and there is no such thing as a noiseless channel.
2. If C > h, then there is room to encode more bits per signal. Though ideally we want to reduce redundancy, a redundant (longer) encoding can better identity a message through a noisy channel.

Lecture 35

1. h = -log(1/10)
2. Because the frequency of symbols are always changing. We can parse a dictionary and calculate the probabilities of alphabets, diagrams, trigrams, but a dictionary doesn't reflect word usage in the english language. Therefore, you would have to calculate and parse many different sources of text in order to have an accurate measure of probabilities. In addition, you have to parse many different varieties of sources, so your probabilities aren't skewed to a particular subject or topic.
3. Zero order: all symbols are equally likely. First order: symbols are not equally likely, but are independent of each other. Second order: computes probability based off of digrams, or 2 particular symbols occurring one after the other. Third order: Like, second order, but 3 particular symbols.

Lecture 36

1. Because information content depends on the state/knowledge of the receiver, who may not be in the same state or the same knowledge as other receivers.
2. The information content means differently to observers in different states, despite being the same message. Consider the example of "the attack is at dawn" in the last homework assignment.
3. Entropy measures redundancy. If information content of a message is equal to the length of the encoded message, there is no redundancy and the encoding is efficient. For instance, my mom is a slow texter, we both know this so she only texts needed words. However, my friends, who love texting, types a lot of unnecessary symbols, such as ellipses…, which isn't necessary to the message.

Lecture 37

1.

underlying language of plaintext? English
characteristics? directional words, N, S, E, W, number of paces
source language - alphabet frequency in english language
nature/complexity - simple algorithm, simple substitution perhaps
transformations/compressions - maybe the message squeezed out all the spaces or vowels.

2. Keys may be optional because encryption and decryption may use the same function.
3. Information content is preserved, only hidden or obscured so that only the intended reader can understand it.
4. Redundancy in plaintext is reflected in the cipher text and can be leveraged for decryption.

Lecture 38

1. Rewrite the following in its simplest form: D(E(D(E(P)))).

E(P) = C
D(C) = P
E(P) = C

D(C) = P

P

2. Rewrite the following in its simplest form: D(E(E(P, Ke), Ke), Kd).

E(P, Ke) = Ce
E(C, Ke) = Cee
D(Cee, Kd) = Ce

C
Assuming that Kd and Ke are the same, symmetric.

3. Why might a cryptanalyst want to recognize patterns in encrypted messages?

Traffic analysis: a cryptanalyst might not be able to decrypt the message, but if they observe a familiar stream of messages due to a certain scenario, it might give them a clue about what is going on. While they might not have broken that algorithm, they may be able to infer some meaning.

In addition, they may use these patterns to find weaknesses in the implementation of the encryption so they may exploit it.

4. How might properties of language be of use to a cryptanalyst?

The frequency of symbols in the English language may help a cryptanalyst find a fitting decryption algorithm. A cryptanalyst may analyze at different *-order models in addition to individual alphabets.

Lecture 39

1. Explain why an encryption algorithm, while breakable, may not be feasible to break?

An encryption algorithm is breakable if, given enough time and data, an analyst can recover the plain text. However, it may not be feasible to break because there may be too many keys to try without raising suspicion. In addition, the analyst must be able to recognize when they have succeeded, meaning they would need some plaintext/ciphertext pairs to compare to.

2. Why, given a small number of plaintext/ciphertext pairs encrypted under key K, can K be recovered by exhaustive search in an expected time on the order of 2^{n-1} operations?

If there is a keyspace/bit string of size n, there are 2^{n-1} possible combinations of bit strings that can be tried and recovered. If the bit string is size 2, then the exhaustive search would involve checking ’00’, ’01’, ’10’, ’11’. On average, it only takes 2^{2-1} =2 operations to find the right key.

3. Explain why substitution and transposition are both important in ciphers.

A single substation or transposition encryption may seen very ineffective. However, almost all modern commercial symmetric ciphers use combinations of substitution and transposition, often multiple times.

4. Explain the difference between confusion and diffusion.

Confusion replaces the original symbol with another symbol, giving the attacker trouble extracting the original symbol from the encrypted symbol. If an attacker had a plaintext/ciphertext pairing for the

symbol and cipher symbol, they will be able to extract the original symbol. Diffusion rearranges or distributes information widely over cipher text in a way such that the plaintext is not in its original spot.

5. Is confusion or diffusion better for encryption?

I think that diffusion is better for encryption because I'm sure there are already a lot of plaintext/ ciphertext pairings available to attempt to decrypt information.

Lecture 40

1. What is the difference between mono alphabetic and polyalphabetic substitution?

A monoalphabetic substitution has a 1 to 1 mapping from one alphabet to another.

A polyalphabetic substitution replaces a plaintext symbol with another symbol depending on where the plaintext symbol is. That is, a plaintext symbol may map to one or more cipher symbol.

2. What is the key in a simple substitution cipher?

The key in a simple substitution cipher is another symbol in the language, the one that the plaintext is mapped to uniquely.

3. Why are there k! mappings from plaintext to cipher text alphabets in simple substitution?

Because there is only a 1-1 mapping, there are only k! mappings.

In the english alphabet, if a maps to another letter, than b can only map to 25/26 letters remaining, and c can only map to 24/26. Thus, for the english alphabet, there are 26! mappings from plaintext to cipher text.

4. What is the key in the Caesar Cipher example?

The key is the shift distance i such that the key for a symbol x has a key of the alphabet at x + i.

5. What is the size of the keyspace in the Caesar Cipher example?

25 or 26 depending on how you look at it.

6. Is the Caesar Cipher algorithm strong?

No. You probably don't have to try all the keys to get the right one.

7. What is the corresponding decryption algorithm to the Vigenere cipher text example?

Given a Vigenere table V, suppose the cipher text symbol is R and the key is M. Then the plaintext symbol is x such that V[x][M] = R. .

Lecture 41

1. Why are there 17576 possible decryptions for the "xyy" encoding on slide 3?

Since "xyy" encodes a string in the English alphabet, there are 26 possible substitutions per plaintext letter.

26 * 26 * 26 = 17576 different combinations of 3-lettered decryptions.

2. Why is the search space for question 2 on slide 3 reduced by a factor of 27?

A simple substitution cipher has a 1 to 1 mapping between symbols, and there are two distinct symbols in "xyy" Therefore, if there are 26 possibilities for 'x', then there are only 25 possibilities for 'y'. Therefore there are only

26 * 25 different combinations of 3-lettered decryptions, where the last two letters of the decryption are the same.

3. Do you think a perfect cipher is possible? Why or why not?

I think so. If a encryption is more and more dynamic, then I think there is a chance that a perfect cipher would exist.

Lecture 42

1. Explain why the one-time pad offers perfect encryption.

Even if an attacker knew of the cipher text and the presence of the one-time pad, every possible plaintext can be the pre-image of the cipher text with a working key. In other words, because of the many to many relation between the plaintext and ciphertext, you still have to try all the possibilities.

2. Why is it important that the key in a one-time pad be random?

The key has to be random because you shouldn't be able to predict or observe a trend in the production of the keys or else an attacker can design an algorithm to parallel the key production.

3. Explain the key distribution problem.

In order for the receiver to decrypt information from the sender, the sender has to send a key to the receiver to interpret the encrypted information. However, the channel in which they send the keys must be secure, or else an attacker can intercept the plaintext or the key. At the same time, if the communication channel was secure, there wouldn't be a need to have a key or encrypt the plaintext.

Lecture 43

1. What is the downside to using encryption by transposition?

Transposition preserves the symbols of the text, therefore, you can do statistical analysis on the frequency of the symbols to determine that it is in fact a transportation encryption. For an attacker, this reduces the amount of techniques they have to try, which is not what we want.

Lecture 44

1. Is a one-time pad a symmetric of asymmetric algorithm?
Symmetric. The key is used to encrypt and decrypt information.

2. Describe the difference between key distribution and key management.

Key management involves preserving the safety and availability of a set of keys. In other words, how do I get a key from one place to the other safely?

Key distribution involves making sure that the keys we distribute to establish connection are secure.

3. If someone gets a hold of Ks, can he or she decrypt S's encrypted messages? Why or why not?

No. If someone gets a hold of Ks, they need to be get the private key to decrypt S's encrypted messages.

4. Are symmetric encryption systems or public key systems better?

I think each encryption system have their own strengths. Public key systems help reduce or eliminate key distribution problems, but are expensive to generate, and have special structures that may be observed by attackers. Symmetric encryption systems have key distribution problems, but the management of the keys are pretty intuitive. However, there is a $O(n^2)$ relation for symmetric keys, where n is the number of users.

Lecture 45

1. Why do you suppose most modern symmetric encryption algorithms are block ciphers?

I think most modern information is stored throughout systems, therefore it would be difficult to store bitwise encryptions. In addition, modern block-structured ciphers are non-malleable, which is important.

2. What is the significance of malleability?

Malleability is a bad thing for an encryption algorithm. It means that you can make predictive changes in the plaintext just by altering the cipher text.

3. What is the significance of homomorphic encryption?

Homomorphic encryption is the when cypher text of a data that is still usable, as if the original information had not been encrypted at all. This allows for algorithms to run on encrypted text without compromising the encryption.

Lecture 46

1.Which of the 4 steps in AES uses confusion and how is it done?

subByte - replaces a byte in the array replace it by another byte in a table.
mixColumns - multiply state by a fixed 4x4 matrix of integers.
addRoundKey - XOR the state with a 128-bit round key derived from the original key K by a recursive process.

2. Which of the 4 steps in AES uses diffusion and how is it done?

shiftRows - shift the first row by 1, second row by 2, 3rd row by 3

3. Why does decryption is AES take longer than encryption?

When inverting the MixColumns step, the algorithm has to multiply each column by a fixed array before it can continue.

4. Describe the use of blocks and rounds in AES.

It is the repeated use of algorithms on a block of information. These operations further encrypt the data. Each round is different because the state of the block and the key is different.

5. Why would one want to increase the total number of rounds in AES?

Increasing the total number of founts in AES will further "mangle" the data, increasing the steps necessary to decrypt it, and therefore increasing the time it will take to decrypt it, as the decryption algorithm is the inverse of encryption.

Lecture 47

1. What is a disadvantage in using ECB mode?

If you have the same plaintext block, then you have the same ciphertext block.

2. How can this flaw be fixed?

Cipher Block Chaining - you want to do something to randomize the plaintext blocks before they begin encryption. CBC works by XOR each successive plaintext block and then encrypt. Therefore, your XOR is unique because of the position of the plaintext block.

3. What are potential weaknesses of CBC?

You can observe changes to the cipher text over time and backtrack to where the first block changed.

If you find two identical cipher text blocks, then you can derive an equation such that

$$C_{i-1} \oplus C_{j-1} = P_i \oplus P_j,$$

In other words, you'd be able to derive the relation between two plain texts from two cipher texts.

4. How is key stream generation different from standard block encryption modes?

Key stream generation modes use encryption algorithms to generate random appearing stream of bits in a reproducible fashion. The output appears random, but you can reproduce the plaintext by knowing the input and algorithm. A key stream generation would appear more random than a standard block encryption, which is prone to patterns.

Lecture 48

1. For public key systems, what must be kept secret in order to ensure secrecy?

The private key, or the decryption key, must be kept secret because knowledge of the private key and access to the public key allows for decryption.

2. Why are one-way functions critical to public key systems?
One-way functions are critical to public key systems because if you can correctly calculate the inverse of the key, then you're able to decrypt encrypted information.

3. How do public key systems largely solve the key distribution problem?

A public key doesn't need to be sent through a secure channel. Therefore, if two people want to open communication, they would just share the public keys and decrypt using their private keys.

4. Simplify the following according to RSA rules: {{{P}K−1}K}K−1.

{{P}k -1} = C
{{P}k-1}K = P
{{{P}k-1}k}k-1} = C

C

5. Compare the efficiency of asymmetric algorithms and symmetric algorithms.

Symmetric algorithms use machine-level operations, bitwise, arithmetic, which are very efficient to implement.

Public keys algorithms require more complicated computations, such as factoring.

Therefore, public key encryption takes about 10,000 times longer to encrypt a message compared to a symmetric encryption.

Lecture 49

1. If one generated new RSA keys and switched the public and private keys, would the algorithm still work? Why or why not?

Yes, because the keys behave in a symmetric (complementing) fashion.

{{P}d}e = P = {{P}e}d

2. Explain the role of prime numbers in RSA.

The RSA algorithm relies on factoring large numbers. The large numbers used are the products of prime numbers.

3. Is RSA breakable?

Yes

4. Why can no one intercepting {M}Ka read the message?

Because only A has the private key, which decrypts the message encrypted by the public key Ka

5. Why can't A be sure {M}Ka came from B?

Because anyone can have A's public key.

6. Why is A sure {M}Kb^-1 originated with B?

Because only B has its private key, which it sent to A.

7. How can someone intercepting {M}Kb^-1 read the message?

Because they can use B's public key to decrypt it.

8. How can B ensure authentication as well as confidentiality when sending a message to A?

You need two sets of keys, one for authentication/signing, and another one for privacy.

Lecture 50

1. Why is it necessary for a hash function to be easy to compute for any given data?

Because hash functions need to be computed often by different users in order to assure integrity.

2. What is the key difference between strong and weak collision resistance of a hash function?

A strong collision resistant function means it is hard to find any two messages that have the same hash value whereas a weak collision resistant function means that, given a message, it is hard to find another message that hashes to the same value.

3. What is the difference between preimage resistance and second pre image resistance?

Preimage resistance means that it is hard to find a message with only a given hash value whereas in second pre image resistance, given a message, it is hard to find another message that share the same hash value.

4. What are the implications of the birthday attack on a 128 bit hash value?

For a 128 bit hash value, on average, you have to look at 1.25 * 2 ^64 values before you find a collision.

5. What are the implications of the birthday attack on a 160 bit hash value?

For a 160 bit hash value, on average, you have to look at 1.5 * 2 ^80 values before you find a collision.

6. Why aren't cryptographic hash functions used for confidentiality?

Because confidentiality isn't concerned with the contents of a file, and cryptographic hash functions are used to make sure that if you are trying to access a message that should have a hash value H that you are not getting the message that computes to a different hash value.

7. What attribute of cryptographic hash functions ensures that message M is bound to H(M), and therefore temper-resistant?

Because the characteristics of M are what yields a hash value, any changes in M would compute to an ideally different hash value. Therefore you can compare the hash values of messages when you are trying to read/write to see if it has been tampered with.

8. Using RSA and a cryptographic hash function, how can B securely send a message to A and guarantee both confidentiality and integrity?

Using RSA, B can send a message to A encrypted by A's public key, and A can decrypt the message with its private key. This allows only A to understand the message that B sent, ensuring confidentiality. B can also send a cryptographic hash function to A along with the message. When A gets the message, A computes the hash value of the message. If the computed hash value does not match the one that B sent, then the message has been tampered with.

Lecture 51

1. No. In this message, S is sending a message to R, first encased in R's private key. Any attacker can R's public key to remove this layer of encryption, revealing the message encrypted by S's private key.

The second layer is encrypted in S's private key, and any attacker again can use S's public key to decrypt and remove that layer. Then the message is decrypted.

2. No. In this message, S is sending a message to R, first encrypted by S's private key. An attacker can remove this level of encryption by using S's public key, revealing the message encrypted by R's public key. R will be the only one that is able to read the message since only R has it's private key, but the authentication layer that guarantees that S is the sender is gone.
3. No
4. Requirements of a key exchange are confidentiality and authentication. Confidentiality ensures that the receiver is the only one that can decrypt the information, and authentication ensures that the sender must find some way to let the receiver know that it really is them sending the message.

Lecture 52

1. Nothing, because you don't know b, so you can't find the key. Rather, it would take forever.
2. Nothing, because you don't know b, so you can't find the key value.
3. Nothing, because you don't know a, so you can't find the key value.