Lucas Harrison
lmh2538
lucash
harrisonlucas@utexas.edu

Lecture 34

1. Why is it impossible to transmit a signal over a channel at an average rate greater than C/h?

Because C represents the overall capacity of the channel and h represents the entropy of the language. C/h therefore represents the best case scenario of an average rate of transfer. The capacity is maxed out and the number of bits per symbol is arbitrarily close to entropy on average.

2. How can increasing the redundancy of the coding scheme increase the reliability of transmitting a message over a noisy channel?

Suppose you had an encoding scheme representing yes, no, and maybe. If yes was 6 1's, no was 6 0's, and maybe was 101010, the encoding scheme is super redundant. However, If on average two or three of these bits are lost to noise, then the original message is still recoverable since they are so distinct and long enough that you can be relatively certain you have the correct message.

Lecture 35

1. If we want to transmit a sequence of the digits 0-9. According to the zeroorder model, what is the entropy of the language?

h = -(log(1/9)) since all the digits are equally likely.

2. What are reasons why computing the entropy of a natural language is difficult?

It is very difficult because it requires super sophisticated models that can handle high order modeling of the language as a whole. Also, it would be difficult in itself to determine a good enough sample size that represents the languages usage on average. You couldn't use every book ever for data. Instead, you would have to pick ones from this century and across various genres and reader levels to really grasp the normal usage of a language.

3. Explain the difference between zero, first, second and third-order models.

Zero and one both assume that each symbol is independent. Zero doesn't take into account the probablity of said symbol. One order model steps it up and takes the probability that a single symbol will occur into account and can lower the entropy by doing so. Two and three use the probabilities that a certain symbol will accompany another symbol to lower the entropy further.

Lecture 36

1. Why are prior probabilities sometimes impossible to compute?

2. Why is the information content of a message relative to the state of knowledge of an observer?

3. Explain the relationship between entropy and redundancy.

The closer the efficiency of a message is to the entropy, the less redundant a message is. If the efficiency matches the entropy, then thats as efficient as a mesage can be represented and therefore there is no redundancy.

Lecture 37

1. List your observations along with their relevance to cryptography about Captain Kidd's encrypted message.

The setting is important. Knowing that it was sent from a pirate means that it may be old/slang english and might have nautical terms. Inferring its directions to treasure could make more navigational terms likely. Also, assuming its English makes counting the symbols and associating them based on probability of a letter is a solid approach.

2. Explain why a key may be optional for the processes of encryption or decryption.

There could be an algorithm that encrypts and decrypts based on the previous letter. So the first time an 'e' appears, it might have a different symbol representation than the second time. This way does not involve a key but still encrypts the information.

3. What effect does encrypting a file have on its information content?

Assuming there is no errors with encoding/decoding, the information contained in the original message is uneffected by the encryption process. It can condense the message perhaps or even make it longer if the encoding scheme is overly redundant.

4. How can redundancy in the source give clues to the decoding process?

If you know something about the underlying language or the source of the message or its probable message, you can use that to make educated guesses on what the most common symbols might represent. For example, if you know it is an english message and one symbol occurs most, that symbol is likely to represent an 'e' or another common character.

Lecture 38

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

P. It's just P because plain text = D(E(P)) and if you replace that with P you get, D(E(P)) which is just P again.

2. Rewrite the following in its simplest form: $D(E(E(P,K_E),K_E),K_D)$.

Uhhhhhhhh, not sure about this one. It seems like this is trying to encrypt a message twice? If I apply C = E(P,KE), it comes to D(E(C),KE),KD). I don't see how encrypting cipher text is a good idea...

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

Patterns could be associated with common patterns in the original message's language. You could infer overall situational information based on the number of occurences of patterns. For example you could guess that there's a fire if a fire station's encrypted message output doubled breifly.

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

The probability of characters or symbols can be helpful in determining the decoding key algorithm. Knowing the source helps you further imply more about how the language might be being used.

Lecture 39

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

There could be so many possibilities to brute force through that there is not a computer fast enough that would break the encryption in the foreseeable future.

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

Because if the same key K is being used you don't have to try that key more than once.

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

They are the building blocks for many modern commercial sipher operations. By changing the symbols value and position, you can effectively confuse and diffuse the attacker.

4. Explain the difference between confusion and diffusion.

Confusion is basically subsitution. When subsituting a symbol for another one, you are changing the information from the plaintext to the cipher text in a way that can't be easily extracted. Diffusion is moving or stretching/condensing a region of plaintext (could be a word or sentence or anything) widely over the ciphertext.

5. Is confusion or diffusion better for encryption?

Confusion is better because with only diffusion, the encryption algorithm isn't too strong. Piecing together a language you already understand isn't as difficult as deciphering a bunch of seemingly random symbols.

Lecture 40

1. What is the difference between monoalphabetic and polyalphabetic substitution?

Monoalphabetic is simple substitution. There exists a 1-1mapping from plain to

ciphertext. This makes it a not very strong encryption.

2. What is the key in a simple substitution cipher?
Since its a 1-1 mapping, it could be anything you chose. The key just describes which symbols are subsituted for which symbols in the plain text.

3. Why are there k! mappings from plaintext to ciphertext alphabets in simple substitution?
Where k is the size of the alphabet there are only k! possibilities because one of those is the correct 1-1 mapping.

4. What is the key in the Caesar Cipher example?
The key is the fixed distance the symbols are replaced by. So if it were three, A would be replaced by D and B would be replaced by E and so on.

5. What is the size of the keyspace in the Caesar Cipher example?
If we're using english, it would be 26.

6. Is the Caesar Cipher algorithm strong?
Not really because you could guess and check two symbols with different key values until you find a match that works from the cipher text.

7. What is the corresponding decryption algorithm to the Vigenere ciphertext example?
If you have the key and the cipher text the decryption algorithm is to go to the row of the key letter and then find the ciphertext letter in that row and then the column that its in is the plain text letter. For example, go to row M, look over 6 spaces and see the letter 'r', then look up and youre in column F which is the correct plain text letter.

Lecture 41

1. Why are there 17576 possible decryptions for the “xyy” encoding on slide 3?
Because each of the three letters can correspond to different three letters since its a substitution cipher but not necessarily simple substitution. Therefore, the x can represent 1 of 26 letters, the first y can represent 1 of 26 letters, and so can the second y leading to 26 * 26 *26.

2. Why is the search space for question 2 on slide 3 reduced by a factor of 27?
Since its a simple subsitution, x can only correspond to one other symbol and so can y. That means both y's correspond to the same letter. Therefore x represents 1 of 26 possibilities and y represents 1 of the remaining 25.

3. Do you think a perfect cipher is possible? Why or why not?
I imagine that one exists today. It would make sense if there were a very good encryption algorithm that is used widely where the best option for decryption without the key is brute force. That way companies wouldn't care if their encrypted messages were found because they're confident the algorithm will hold.

Lecture 42

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

Because even though you know both the cipher text and the algorithm, you can't reduce the possible plain text answers that it could represent. Knowing the cipher is 101 doesn't reduce any of the 8 possible choices that it might represent.

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

If you know anything about the key you might be able to recognize patterns or as the slides showed if you knew that it was an even parity key then you can reduce the possible plain text representations by half which makes it a non-perfect encryption.

3. Explain the key distribution problem.

Having a long key like in one-time pad requires the sender and reciever to have to agree on a key. The key distribution problem comes up when trying to get the key from sender to reciever. If you have a secure channel to do this, then why even bother with the key? If not what are your options?

Lecture 43

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

The frequencies of the plain text are preserved in the cipher text so if used by itself, you give the attacker a lot to work with when attempting to decrypt your ciphertext.

Lecture 44

1. Is a one-time pad a symmetric or asymmetric algorithm?

It is a symmetric algorithm because the key is applied to the plain text to produce the ciphertext and visa versa.

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

Key distribution is the problem that arises when trying to convey a key from sender to reciever securely. Key management is as the keys start to pile up, how do we preserve their safety while still keeping them available when needed.

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

Nope. You would need the decryption key. Ks is even called S's 'public key' because it doesn't matter who can see it. Only the Ks-1 can decrypt S's messages.

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

Asymmetric encryption systems (public key systems) seem much better since they address and fix the key distribution problem since the key is now public and their is no need for the sender to also send the key with its encrypted message.

Lecture 45

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

The main advantage I believe is the high amount of diffusion you get with block algorithms. Obtaining a block of cipher text doesn't really tip you off to anything about the original plain text. However, in a stream encryption, if you can see several symbols of the cipher text you might be able to extract patterns and gain some possible meaning about the original plain text.
\
2. What is the significance of malleability?

If an encryption is malleable, attackers can use that to potentially corrupt and change the original message from the plain text without the end user ever noticing. Non-malleability is important because that way tampering with the cipher text will entirely ruin the message the reciever sees which alerts them to an error or attack.

3. What is the significance of homomorphic encryption?

Homomorphic encryptions seem pretty good in that they handle key distribution completely. The sender uses one algebraic function to encrypt and the reciever uses another algebraic function to decrypt. Also, key management would be easier since you could create tons of different algebraic functions but it may be difficult to create a pair that work together.

Lecture 46

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

subBytes. This step in each round substitutes each byte in the array using its value as an index into a 256 element lookup table which replaces the byte with the value in the table. addRoundKey XOR's the state which affects the original key and thus adds to the confusion.

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

shiftRows. This step shifts the row by various numbers of bytes diffusing the original message. Also mixColumns helps diffuse the origninal message by replacing the state's column value by multiplying a fixed number and assigining a new column accordingly.

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

Because during the mixColumns stage on decryption, you have to multiply by the inverse of the original 4x4 matrix. For example, if the original multiplied value was one, its very easy to optimize since you don't have to do anything. Or if the original was 2 you could optimize with shift. The inverse, however, is not easy to optimize and takes longer.

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

AES uses 16 byte blocks (128 bits) and then applies simple operations that perform substitution and transformation on the bytes. That is one round. Then, these operations are repeated on the new 'state' of the block another 10+ times in order to completely mangle the original message.

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

By adding more round to AES, you can increase the key space which makes the possible key combinations increase making it even harder to brute force the solution.

Lecture 47

1. What is a disadvantage in using ECB mode?

The main disadvantage is that identical blocks in the plain text form identical blocks in the cipher text. This doesn't do a good job of hiding patterns and a lot of info can be learned by analyzing the cipher text.

2. How can this flaw be fixed?

You need to randomize the blocks before encrypting. One way is cipher block chaining.

3. What are potential weaknesses of CBC?

Observed changes and content leak. If the attacker can find two identical ciphertext blocks then they can derive information about the plaintext blocks.

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

The key stream generation is used to generate random appearing streams of bits in a reproducible fasion. It differs in that the encryption isn't actually random as if you were doing random block encryption.

Lecture 48

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

The private key must be secure or otherwise anyone can decrypt your message.

2. Why are one-way functions critical to public key systems?

Because you can release the end result of your one way function and be confident there is no reliable way to get the original inputs to that function. These are the public keys.

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

Having the encryption key public solves the secrecy problem. The reciever already has the decryption key (private key) and therefore there doesn't need to be some secure channel in which to deliver the key from sender to receiver.

4. Simplify the following according to RSA rules: $\{\{\{P\}_{K^{-1}}\}_K\}_{K^{-1}}$.

P k-1.

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

With asymmetric algorithms, the keys get more and more complex involving one way functions and complex operations. This makes them much more time consuming to perform encryption on. Symmetric algorithms on the other hand use very simple operations that can be optimized at the hardware level and therefore remain much faster at encryption/decryption than their asymmetric counter parts.

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 it would still work because you can use either the public or private key for encryption and then the other for decryption.

2. Explain the role of prime numbers in RSA.

The idea that the product of two primes produces a large number that is very difficult to factor to the original two primes. This makes the public key (the product) secure even though anyone can see it.

3. Is RSA breakable?

Yes, you could keep factoring and trying every possible two prime numbers to see whether they are the correct key or not but the idea is that this is extremely time inefficient.

4. Why can no one intercepting $\{M\}_{K_a}$ read the message?

Because no one except the desired recipient has the private key to A.

5. Why can't A be sure $\{M\}_{K_a}$ came from B?

Because since the encryption key is public, anyone could write a message and encrypt it with ka, then send it to the recepient and they would have no assurance it actually came from B.

6. Why is A sure $\{M\}_{K_{1b}}$ originated with B?

Because no one else has B's private key. So knowing that its encrypted correctly with B's private key, A can be assured it originated from B but it has the downside that it can be read by anyone using B's public key.

7. How can someone intercepting $\{M\}_{K_{1b}}$ read the message?

By using B's public key to decrypt the message since {{P}d}e=P={{P}e}d.

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

You can't have both using the RSA public key system.

Lecture 50

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

To make sure that the encryption/decryption stages don't become too time consuming.

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

Strong collision resistance is when its hard to find any two messages that have the same hash value. Weak collision resistance is when you have a message and its hash and it is hard to find another message that produces that hash.

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

Preimage resistant is when given the hash, it is hard to find the message that produced the hash. Second preimage resistance is when given the message, it is hard to find another message that produces the same hash.

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

It means that on average you'd have to look at 2^64 * 1.25 values before you find a collision. Thats a pretty large set.

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

Same as above except even larger set. It'd be a set of 2^80 *1.25 values before finding a collision on average.

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

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

The properties of a cryptographic hash make it so any changes to the file are very apparent since the chances of the change having the same hash are extremely small given a good hash implimentation.

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

If B sends the message with its private key it provides integrity (B really did send this message). In addition to this, B would have to encrypt the message using a cryptographic hash. This works because even though anyone can tamper with the message using B's public key, when A recomputes the hash and compares it to the stored value, then A can be almost completely certain no tampering happened if the values match.

Lecture 51

1. For key exchange, if S wants to send key K to R, can S send the following message: $\{\{K\}_{K_{S^{-1}}}\}_{K_{R^{-1}}}$? Why or why not?

No because anyone can then decrypt the original message using the sender's and receiver's public keys.

2. In the third attempt at key exchange on slide 5, could S have done the encryptions in the other order? Why or why not?

Yes because you could decrypt the outer layer using the sender's public key but then you would still need the receiver's private key to decrypt the actual message.

3. Is $\{\{\{K\}_{K_{S^{-1}}}\}_{K_R}\}_{K_S}$ equivalent to $\{\{K\}_{K_{S^{-1}}}\}_{K_R}$?

No because the reciever would need the sender's private key in order to access the message.

4. What are the requirements of key exchange and why?

It needs to be confidential and authenticated. Both parties need to know that no one has tampered with the message and that it did indeed come from who it says it came from.

Lecture 52

1. What would happen if g, p and $g_a$modp were known by an eavesdropper listening in on a Diffie-Hellman exchange?

Nothing because b could still be any number. It doesn't narrow down the key space by any.

2. What would happen if a were discovered by an eavesdropper listening in on a Diffie-Hellman exchange?

Nothing because they'd still need quite a few other numbers in order to compute the secret number.

3. What would happen if b were discovered by an eavesdropper listening in on a Diffie-Hellman exchange?

Still nothing because a and b are just arbitrary numbers chosen by the two communicators.