WEEK 3 QUESTIONS

Name: Charu Sharma
EID: cs36739
CS Login: charu
E-mail: charu.sharma@utexas.edu

LECTURE 34:

1. A perfect channel achieves perfect entropy,, so the number of bits divided by the entropy should give you the optimal C/h transmission rate, as you get arbitrarily close to the entropy of the encoding system.
2. A redundant encoding system means that a message if lost once on a noisy channel will more likely be caught when sent redundantly afterwards. Therefore a redundant transmission is a more reliable transmission system over a noisy channel.

LECTURE 35:

1. H= -(log 1/10)
2. Not all symbols in a language are equally likely, so frequency must be accumulated to compute prior probabilities before computing entropy with a higher order model. Additionally, the ordering of letters is not entirely random. Therefore, we must implement probabilities that one letter will fall after another. These factors in computation complicate the process of finding the entropy of the language.
3. A zero-order model assumes that each symbol in a language is equally likely and computes the entropy accordingly. A first-order model uses prior probabilities of each individual character in a language, and it uses these prior probabilities to more accurately compute an entropy value. A second-order model considers bigrams, computing entropy values considering the probability that the second symbol will come after the first in a bigram. A third-order model considers trigrams, computing entropy values considering the probabilities that the second symbol will come after the first in a trigram and the probability that the third symbol will come after the second in a trigram.

LECTURE 36

1. In actuality, entropy is relative to a particular observer, since it depends on how much the observer knows about the scenario. For this reason, prior probabilities are often impossible to compute.
2. Each observer may know different amounts about the likelihood of events in a scenario. For this reason, prior probabilities would change based on these likelihoods. For instance, at the Academy Awards, the prior probabilities would change depending on how much a given observer knew and understood about favored winners.
3. The closer a language is to its entropy, the less redundant its encoding is. If the information content of the message is equal to the length of the encoded message, there is no redundancy.

LECTURE 37

1. One might ask what the underlying language is, which is important because our decoding of the language will depend on mapping it to this underlying language. The discovery of the goat's head with Captain "Kidd" points to the high probability of this language being English. We then look for characteristics of the probable source texts

that are relevant; in this case, we might look for directional words since it is a treasure hunting document. We then look for characteristics of the source language which are relevant, namely common letters, etc. so that we can decode the language more easily, mapping the relevant source language characters to relevant encoding language characters. Then we can look at the likely nature/complexity of the encryption algorithm to know how hard we'll have to work. For instance, in this case, the work having been done by a pirate is probably something simple like simple substitution. We can then look at what transformations or compressions have been applied, so we know whether it is a one to one mapping or not.

2. The encryption algorithm could be enough to turn plaintext into ciphertext. A key can just help in this process, whether it is the same or different for encryption and decryption.
3. The information of the message should be preserved, otherwise the message won't be recoverable or comprehensible to the receiver.
4. The redundancy can imply regularities about the source code or language of the encrypted code. Therefore, redundancy can help an attacker more easily decrypt code by using redundancies to match up regularities of source code. It is the enemy of secure encryption since it provides leverage to the attacker.

LECTURE 38

1. The simplest form is just P.
2. E(P, Ke) is the simplest form.
3. Recognizing patterns in an encrypted message also maps to recognized patterns in the source language, helping the decryption process. Additionally, traffic analysis can use patterns to help encryption of a language as well.
4. Frequency of characters for instance can help map an encrypted code to a source language and certain characters of the source language as well. Properties of a language can help determine the source language, and then subsequently use properties of the encryption as properties of the source language to more easily decode the encrypted message.

LECTURE 39

1. An encryption algorithm is breakable if given enough time and data, an analyst can recover the plaintext from the cipher text. Most encryptions are breakable, because the analyst can easily try all keys systematically. However, breaking the ciphertext can take a long time, making it unfeasible. Additionally, the analyst will probably not be able to recognize success, and unless you have a plaintext/ciphertext pair or multiple that are known to match, most ciphertexts aren't easily or feasibly breakable.
2. There are 2^n possibilities for a n-bit string as a key. Additionally, a linear search through these possibilities requires half the operations, so you get 2^n-1 operations to search.
3. Substitution is when each symbol is exchanged for another, though not necessarily uniformly. Transposition is when the order of symbols is rearranged. Substitution offers an easily encoded message, since symbols are just switched out for one another. Transposition offers a more difficultly decoded message, since the order of symbols has been changed. When used together, substitution and transposition can form an effective encoding system.

4. Confusion is transforming information in plaintext so that an interceptor can't readily extract it, and is accomplished through substitution more than through transposition. Diffusion is the spreading of information from a region of plaintext widely over the ciphertext, and is established more through transposition than through substitution.
5. Both confusion and diffusion are good and necessary for symmetric encryption algorithms.

LECTURE 40

1. Monoalphabetic substitution is the uniform substitution of characters, but a polyalphabetic substitution is a non-uniform substitution of characters, meaning that different substitutions are made depending on where in the plaintext the symbol occurs.
2. In the simple substitution cipher, the key is however you specify the 1-1 mapping of an alphabet into itself or another alphabet. It may be contained in a table.
3. There are k! mappings, because each symbol can be mapped to any of the other k symbols in the alphabet.
4. The key in the Caesar Cipher is the "distance" between the character and the substituted character or the number of positions shifted between characters for encoding.
5. The size of the keyspace is 25 or 26 depending on how you look at it, since there are 26 letters in the English alphabet.
6. The Caesar Cipher algorithm is not strong. You don't have to try all of the possibilities before getting the right one.
7. The corresponding decryption algorithm to the Vigenere ciphertext example are the 26 Caesar Ciphers as found in the Vigenere Tableau table.

LECTURE 41

1. There are 26^3 decryptions possible, meaning there are 17576 possibilities, because although we know it is a substitution cipher, we don't know that it is a simple substitution cipher.
2. We then know it is a simple substitution cipher, and that the last two characters are both y, giving us 26*25 possibilities. (26^3/(26*25))=27, so we have reduced the possibilities by a factor of 27 by learning it is a simple substitution cipher.
3. I don't think a perfect cipher would be one for which no reduction of the search space would be gained from knowing the encryption algorithm and the ciphertext. The attacker's uncertainty of the message in a perfect cipher is exactly the same whether or not he or she had access to the ciphertext. A perfect cipher would be possible, because these requirements are feasibly met with the one time pad algorithm and requirements of randomness.

LECTURE 42

1. The one time pad offers perfect encryption, because a perfect cipher requires as many possible keys as plaintext with the key chosen randomly. The one time pad uses a key that is the same length as the plaintext and uses it only once. That key is XOR-ed with the plaintext. Every possible plaintext could be the pre-image of that ciphertext under a plausible key, so no reduction of the search space is possible.

2. The key in a one time pad must be random, because if you knew something about the key, then you could work backwards to take the ciphertext and eliminate half the possible plaintext using the knowledge, making it an imperfect cipher.
3. The key distribution problem is a problem that occurs because a key as long as the plaintext has to be taken to the other end of the channel, and the sender and receiver both need a secure channel to send the key, but if the secure channel exists, the plaintext can be sent over that secure channel, and if there isn't a secure channel, it is hard to get the key from the sender to the receiver secretly.

LECTURE 43

1. If you use encryption by transposition, the downside is that even though transposition reorders characters, the original characters still occur in the result, so letter frequencies would be preserved in the ciphertext, though frequencies of digrams, trigrams, etc. wouldn't be.

LECURE 44

1. A one-time pad is a symmetric key, because both parties have the same key for encoding and decoding.
2. Key distribution is the method by which we convey keys to those who need them to establish secure communication, while key management focuses on preserving the safety of a large number of keys to make them available as needed.
3. No, you need Ks-1 to decrypt it, since it is not the same as Ks, the encryption key.
4. Public key systems are better, because they solve the key distribution problem. A different key is used for encryption and decryption, so that the encryption/decryption becomes more secure. Having a private and public key create a secure channel for information to travel over. Also, it is more efficient, because there are only 2n keys, or possibly 4n, depending on the version of public key encryption. However, it is still only O(n). They *can* be expensive to generate though, while symmetric encryption is simple to generate, since the keys are just randomly generated k-bit strings and have no special properties.

LECTURE 45

1. This is probably because a block cipher can encrypt a group of plaintext symbols as one block, which is easier to use on 64 bit or 128 bit block sizes in modern symmetric encryption algorithms. There is high diffusion since information in one plaintext symbol is deiffused into several ciphertext symbols. Additionally it is immune to tampering, because it is hard to insert symbols into the plaintext. It allows variation in block size, also, for a good amount of flexibility.
2. Malleability occurs when transformations on the ciphertext produce meaningful changes in the plaintext. For this reason, malleability is a bad thing for encryption systems. You won't be able to detect the change in encryption, but the message content may have changed significantly without detection.
3. Homomorphic encryption is a type of encryption in which specific kinds of computations can be carried out on ciphertext and create an encrypted text which matches the result of operations on the plaintext when decrypted.

LECTURE 46

1. Step 1, subBytes introduces confusion into the encryption system by implementing simple substitution, as confusion is often introduced.

2. Step 2, shiftRows, applies transposition into the encryption system, thereby introducing diffusion into the AES system. You shift rows over to introduce transposition.
3. Decryption in AES takes longer than encryption, because inverting the MixColumns step requires multiplying each column by an inverse fixed array, which is irregular and harder to multiply, which makes decryption take longer than encryption.
4. You arrange a set size block into an array of bytes, called states. Those states derived from blocks are modified in place during each round. There are typically 10, 12, or 14 rounds for keys of 128, 192, and 256 bits respectively.
5. You would want to increment the number of rounds in AES, because it makes the encoding more complex, and therefore it makes it harder to decrypt.

LECTURE 47

1. A disadvantage of using ECB mode is that identical blocks in plaintext encode into identical blocks in the resulting ciphertext, making it easier to decode.
2. This flaw can be fixed by having a process through which identical blocks in plaintext don't yield identical blocks in the ciphertext. You can randomize the identical blocks in plaintext by using Cipher Block Chaining (CBC), in which you XOR each successive plaintext block with the ciphertext block and then encrypt with an initialization vector (IV) used as a "seed" for the process initially.
3. One weakness of CBC is that an attacker observing changes to ciphertext over time can spot the first block that changed. Additionally, there is a content leak problem, in that an attacker can find two identical ciphertext blocks, he can derive a relation between them.
4. Key stream generation is used more as a pseudorandom number generator, resulting in a key stream that can be used as a one-time pad. Decryption uses the same key string. On the other hand, block encryption modes generate ciphertext that stores the message in encrypted but recoverable form.

LECTURE 48

1. For public key systems, a secret decryption key must be kept secret to ensure secrecy of a public key encryption system. This private key is K-1.
2. A one-way function is a function which is easy to compute, but difficult to invert without additional information. This is important because it provides an efficient encoding but effectively difficult decryption.
3. Public key systems largely solve the key distribution problem, because the public key can be given without fear of eavesdropping. Now the encryption is public, but decryption is private.
4. $\{P\}_{K-1}$
5. A public key encryption could take up to 10,000 times as long to perform as a symmetric encryption, because the computation depends on more complex operations, not on simple bit-wise operations. For this reason asymmetric encryption is more inefficient than symmetric encryption, making it less common than symmetric encryption.

LECTURE 49

1. Yes it would, because in RSA, an encryption and decryption key can be used in either direction. A public key can be used in encryption or decryption, and the private key

can be used in encryption or decryption, because the keys are used in a somewhat symmetric fashion.

2. The role of prime numbers in RSA is that RSA numbers are semiprimes, or numbers with exactly two prime factors. It is easy to encode this and create that number, but harder to factor it back into primes in its decryption.
3. RSA is not breakable, but it is a privacy transformation, but not an authenticity transformation. You can't be sure where the key is coming from.
4. No one intercepting can read it, because only A has the key that will allow the decryption of B's message, so only he can read it.
5. A can't be sure it originated with B because it is not an authenticity transformation.
6. No one besides B had to private key to send that message.
7. Anybody could have had a public key, so confidentiality could be violated, and a privacy transformation hasn't been secured, and eavesdropping could occur.
8. A public key encryption can be used for authenticity or for privacy but not both at once. You have to use different keys to achieve both.

LECTURE 50

1. A hash function has to be easy to compute because it needs to convert variable sized text into small datum to compress it.
2. A function f is week collision resistant if given an input m1, it is hard to find m2 and m1 which are unequal such that f(m1) and f(m2) are equal. A function f is strong collision resistant, on the other hand, if it is difficult to find two messages m1 and m2 such that f(m1)=f(m2).
3. Preimage resistant functions mean that given h it is difficult to find any m such hat h=f(m), while second preimage resistant says that with input m1 it is hard to find m2 not equal to m1 for which f(m1)=f(m2). This is sometimes called weak collision resistance.
4. The effect of the birthday attack on a 128 bit hash value is that you would have to look at 1.25*2^64 values before you find a collision.
5. The effect of the birthday attack on a 160 bit hash value is that you would have to look at 1.25*2^80 values before you find a collision.
6. Cryptographic hash functions aren't used for confidentiality, because they are more often used for integrity instead which override confidentiality concerns. It binds the bytes of a file together so that any alterations to the file are apparent. In a way we seal the file to make it tamper-resistant, protecting integrity of a file not confidentiality.
7. A cryptographic hash function binds the bytes of a file together in such a way that alterations ot the file are apparent. In this way we seal the file to make it tamper-resistant. The attribute that allows us to do this is the stored hash function result h(f). It's collision resistant, so tampering would be easily detected since the hash values wouldn't probably match.
8. RSA could ensure confidentiality of a file, while the cryptographic has function would guarantee integrity of the file.

LECTURE 51

1. No in that example there is no encryption key, and both keys are decryption keys.
2. No, you can't, because you must first encode the key with S's key and then decode it with R's key, so that no one but R can decrypt the message, ensuring confidentiality,

and no one but S could have performed the inner encryption, ensuring authentication.

3. No, it’s not equivalent.
4. Key exchange requires both confidentiality and authentication.

LECTURE 52

1. The eavesdropper cannot discover the shared secret even knowing that much information, because the individual messages don’t contain the number.
2. Knowing a is not enough, because you can’t track it back to the shared secret.
3. Knowing b is not enough, because you can’t track it back to the shared secret.