Name: Jessica Lucci
EID: jml3624
CS Login: jlucci
Email: jessicalucci14@gmail.com

Lecture 34

1. C/h is equal to the maximum number of symbols per second that can be sent across the channel. Therefore, it's impossible to transmit a signal over a channel at a rate greater than C/h, as it would exceed the channel's physical capabilities.
2. Increasing the redundancy of a coding scheme means that parts of the coding scheme can be omitted without loss of meaning. So by increasing the redundancy, there's less of a chance that parts of the coding scheme will be lost in a detrimental manner in a noisy channel.

Lecture 35

1. h = -(log/10), since each digit 0 - 9 is equally likely.
2. Computing the entropy of a natural language is difficult as letters are not independent of one another (some follow others more frequently, or not at all), and all those possible sets of symbols (digrams, trigrams, etc) occur at varying frequencies.
3. Zero-order models assume every symbol has the same likelihood of occurring, first-order models assume individual symbols are dependent on one another, second-order models assume digrams are dependent on one another, third-order models assume trigrams are dependent on one another, and so forth.

Lecture 36

1. Prior possibilities are often impossible to compute, as entropy is dependent on the observer.
2. Information content is relative to the observer as different observes may have different prior amounts of knowledge about a message. For instance, the judges in a contest already know who the winner is (0 entropy), but the observers of the contest have no idea who the winner is yet.
3. Entropy can be used as a measurement of redundancy in an encoding. For example, if an encoding's efficiency matches the entropy, there is no redundancy present within the language.

Lecture 37

1. Observations:
    a. Frequent use of semicolons - frequency implies that if the semicolon is representative of a single character, the encoded characters is probably used frequently in the underlying language.
    b. Message ends in semicolon - Semicolon could be representative of punctuation

    c. If message is believed to have been sent by Captain Kidd, the underlying language is probably Captain Kidd's native language.
2. A key might be optional, as an encryption algorithm might be defined by the algorithm itself (like the Caesar Cipher).
3. Information content should be preserved throughout encryption. The encrypted message should still resolve the same amount of certainty as it's still sending the original message - just in a different form.
4. Redundancy in the source can give some clues to the decoding process as symbols that tend to be redundant in the source data will most likely have redundant symbol representations in the encrypted data as well.

Lecture 38

1. $D(E(D(E(P)))) = D(E(D(C))) = D(E(P)) = D(C) = P$
2. $D(E(E(P, K_E), K_E), K_D) = D(E(C, K_E), K_D) = D(C^1, K_D) = P^1$
3. Patterns in encrypted messages can often give clues as to what the underlying message is.
4. Properties of a language could help a cryptanalyst to identify patterns in an encrypted language that correspond to the properties in the underlying language. For example, in the english language adjectives come before a noun, where in the spanish language the adjectives follow the noun. So if cryptanalyst was dealing with an underlying english language he'd know to look for descriptive words before nouns, and vice versa if he were dealing with an underlying spanish language.

Lecture 39

1. An encryption algorithm may be breakable, but it may be so complex that it would take massive amounts of computing computing power billions of years to check all the possible keys that would break the code.
2. Many ciphers use an n-bit string as a cipher. There are 2^n ways, then, to organize the bits in this cipher. So for a small number of plaintext/ciphertext pairs encrypted under some key K, a brute force linear search on this 2^n space would feasibly produce a result. Since this is a linear search, it would on average take 2^{n-1} operations to find an appropriate cipher.
3. Substitution and Translation are important in ciphers as almost all modern commercial symmetric ciphers use some combination of substitution and transposition for encryption.
4. Confusion transforms the information contained within plaintext in place, while diffusion spreads the information in the plaintext out across the ciphertext.
5. Both confusion and diffusion are useful tools encryption - both have their own specific advantages, but neither one is explicitly better than the other for encrypting information.

Lecture 40

1. Monoalphabetic substitution is when each symbol in plaintext is exchanged for another symbol, such that there's a 1:1 relation between the symbols in the encrypted text and the symbols in the plaintext (i.e. every e is represented by a 2, etc.). Polyalphabetic substitution is when each symbol in plaintext is exchanged for another symbol such that the substitution is dependent on the symbol location in the plaintext.
2. The key in a simple substitution cipher is however the 1-1 mapping is defined - like a table.
3. There are k! mappings from plaintext to ciphertext alphabets as there are k! ways to rearrange k letters.
4. The key in Caesar Cipher example is something like the number of letters you shift down to pick a mapping.
5. The size of the keyspace is equal to the number of letters in the language. So in English, the keyspace would be 26.
6. The Caesar Cipher is not a strong algorithm - you don't have to try very many possibilities to figure the cipher key out.
7. The decryption algorithm for the Vigenere ciphertext example would be to find all the occurrences of the encrypted letters in the Vigenere Tableau, and check each possible key-pair that could have generated the encrypted letters, until one of the sequences of key-pairs spells out a legible sentence.

Lecture 41

1. There are 17576 possible decryptions for 'xyy', since there are 3 spots with 26 possible symbols for each spot (26^3).
2. The search space is reduced by 27 since we know that 'xyy' is a simple substitution cipher. That means that x will map to a different symbol than y. So once we've determined what x or y represents, we have one less symbol to search for the remaining encoded symbol.
3. I don't believe that a perfect cipher is possible, as all encryptions are reversible. So knowing how the ciphertext was generated (the encryption algorithm) also shows how to reverse the encryption.

Lecture 42

1. The one-time pad theoretically provides perfect encryption as each key is random and only used once, providing no possible reduction of the search space.
2. The key must be random, because if you knew anything about it, you'd be able to eliminate some aspect of the search space. This elimination would take away the perfectness of the cipher.
3. The key distribution problem is the issue of how to send the key to authorized parties securely.

Lecture 43

1. Encryption by transposition has a large space complexity, as the entire encrypted message must be read before it can be decrypted.

Lecture 44

1. One-time pad is a symmetric algorithm as it used the same key for both encryption and decryption.
2. Key distribution is the issue of transporting or distributing keys in a secure manner, and key management is the issue of storing a large number of keys securely.
3. If someone gets a hold of K_S they cannot decrypt K's encrypted messages, because K uses their private key K_S^{-1} to decrypt messages.
4. Neither symmetric encryption or public key systems are better than the other as they each have their own specific advantages.

Lecture 45

1. Most modern symmetric encryption algorithms are block ciphers as block ciphers are very difficult to tamper with - an illegal symbol insertion is usually immediately detected.
2. Malleability is significant in that it allows someone to modify the ciphertext in a legal way that will change the original meaning of the plaintext.
3. Homomorphic encryption is significant in that it allows an algebraic operation performed on the plaintext to equate to an algebraic (possibly different) operation on the ciphertext. This allows for various traits of cryptosystems, such as secureness in a voting poll.

Lecture 46

1. The first step of AES uses confusion by replacing each byte in the array with a value taken from a 256-element look up table.
2. The second step of AES uses diffusion by shifting each of the modified bits by some amount within their respective row.
3. Decryption of AES usually takes longer than encryption as inverting the MixColumns step requires multiplying each column by a fixed 4x4 array.
4. By taking input in fixed size blocks and iteratively performing operations on a 'state', AES is able to process large amounts of data quickly and efficiently (cheap operations such as table look ups, and bitwise operations are used).
5. An increased number of total rounds in AES would create an additional gain in both the confusion and diffusion of the original plaintext.

Lecture 47

1. ECB mode creates blocks that are identical in the ciphertext for blocks that are identical in the plaintext. This can be a problem for plaintexts that have many identical blocks, such as internet packet traffic.

2. The flaw of identical blocks can be fixed by adding an additional factor that will "randomize" blocks before encryption. This will cause identical blocks in the plaintext to appear as different blocks in the ciphertext.
3. An attacker that is able to monitor the ciphertext over a period of time will be able to determine the first block that changed. Additionally, if an attacker is able to find two identical ciphertext blocks, they have access to the relation $C_{i-1} \oplus C_{j-1} = P_i \oplus P_j$, and can gather information about the original plaintext blocks.
4. Standard block encryption modes store the ciphertext in decryptable blocks. Key stream generation uses the cipher as a sort of random number generator instead, that results in a key stream that is essentially equivalent to a one-time pad.

Lecture 48

1. For public key systems, the private key must be kept secret in order to ensure secrecy.
2. One-way functions are essential to public key systems as a one-way function is easy to compute (encryption with public key) but "undoing" that function is extremely difficult (decrypting the message without the private key).
3. Public key systems largely solve the distribution problem as public keys don't need to be protected - you no longer have to worry about securely moving a key around.
4. $\{\{\{P\}_{K-1}\}_K\}_{K-1} = \{P\}_{K-1}$ = private key
5. Public encryption (asymmetric encryption) may take up to 10,000 times as long to perform as symmetric encryption, as public encryption relies on complex operations - not the bitwise operations used by symmetric encryption.

Lecture 49

1. Since $\{\{P\}_d\}_e = P = \{\{P\}_e\}_d$, switching the public and private keys would allow the algorithm to continue appropriately - either key can encrypt or decrypt.
2. Since plaintext is encoded P^e mod n, an interceptor would have to try and factor P^e (a prime number) in order to recover the plaintext.
3. Technically RSA is breakable since it's theoretically possible to brute force search all the prime factors of a number. In reality though, the time needed to brute force search would reach unachievable highs (billions of years).
4. No one intercepting $\{M\}K_a$ could read the message, because they would need A's private key K_{a-1} in order to decrypt the message.
5. A couldn't be sure that $\{M\}K_a$ came from B (or anyone in specific for that matter), because everyone has access to A's public key.
6. A could be sure that $\{M\}K_{b-1}$ came from B, as only B has access to it's private key (K_{b-1}).
7. Someone intercepting $\{M\}K_{b-1}$ could read the message because everyone has access to B's public key.

8. B can ensure authentication and confidentiality by using one key to encrypt the message, and a separate key to "sign" the message.

Lecture 50

1. A hash function must be easy to compute for any given data since the hash function is applied to documents by using the entirety of the document's contents. Documents very often have multiple types of data within them, so a hash function must be able to handle all of those types of data.
2. A hash function has strong collision resistance if it is difficult to find any two messages, m_1 and m_2 such that $f(m_1) = f(m_2)$, while a hash function has weak collision resistance if it is difficult to find any two messages, m_1 and m_2 such that $m_1 \mathrel{!}= m_2 f(m_1) = f(m_2)$.
3. A function is preimage resistant if given *h*, it is difficult to find any *m* such that $h = f(m)$, while a function is second preimage resistant if given input m_1, the function has weak collision resistance (defined above).
4. The implications of the birthday attack on a 128 bit hash value are that you have to look at (on average) $1.25\sqrt{2^{128}}$ values before you find a collision.
5. The implications of the birthday attack on a 160 bit hash value are that you have to look at (on average) $1.25\sqrt{2^{160}}$ values before you find a collision.
6. Hash functions are usually used for integrity instead of confidentiality, as hash functions sort of "bind" the file contents together. If anything in the file is modified, the hash function will produce a different key, and it's immediately evident that the file has been tampered with.
7. Cryptographic hash functions make it virtually impossible to have two unique messages with the same hash. This means that it's easy to make a file "tamper proof" - if the file is changed it will display a different hash than it originally did, and it will become immediately evident that file has been changed.
8. B can use RSA to encrypt and send $\{M\}K_a$ to A to ensure confidentiality. B can then also send a hash of M to A, which A can compare against a hash of its' received version of M to make sure that the file hasn't been tampered with (ensuring integrity).

Lecture 51

1. If S wants to send key K to R, it cannot send $\{\{K\}_{KS-1}\}_{KR-1}$. Since everyone has access to R and S's public keys, anyone could intercept and decode the message using R's public key, and then decode the inner message using S's public key.
2. S could have theoretically sent the message in the opposite order, since no one but R could've decoded the inner message, but anyone could've decoded the outer message which isn't "good" practice.
3. $\{\{\{K\}_{KS-1}\}_{KR}\}_{KS}$ is not equivalent to $\{\{K\}_{KS-1}\}_{KR}$ since R cannot decode the first message's outer most encryption to get to the inner contents (it would need S's private key), while it can get to the inner contents of the second message using its' own private key, and S's public key.

4. Key exchange requires both confidentiality and authentication so that the the receiver knows that the key came from the expected source and has not been tampered with in any way.

Lecture 52

1. If an eavesdropper on a Diffie-Hellman exchange knew g, p and g^a mod p, the eavesdropper could compute the secret key a.
2. If a were discovered by an eavesdropper on a Diffie-Hellman exchange, the eavesdropper would be able to determine the secret key used by both parties (assuming the eavesdropper also knew g and p).
3. If b were discovered by an eavesdropper on a Diffie-Hellman exchange, the eavesdropper would be able to determine the secret key used by both parties (assuming the eavesdropper also knew g and p).