Name: Scott Stephens
EID: sts768
CS Login: scott483
Email: stevo4932@gmail.com

Lecture 36

1. The channel can't handle more than say c = 100 bits/s and language has 5 bits per symbol. Well 5*20=100 bits or 100/5 = 20 letters at a time. Math.

2. by increasing redundancy you increase the number of opportunities for the correct message to get across increases and so does the reliability.

Lecture 35
1. 3.322
2. the probabilities and possibilities are great for natural languages. Not only does frequency of a letter or word play a role but also the ability for humans to fill in missing info.
3. zero is the naive encoding, 1st adds in the frequency of each letter, 2nd adds in freq of a letter after another third loos at frequency of 3 letter combinations.

Lecture 36

1. depends on the likelihood of events which is complicated.
2. observers have different states of knowledge. Someone who knows nothing may need everything spelled out while someone who already knows needs nothing spelled out.
3. entropy can be used to measure the amount of redundancy in the encoding. If info content of a message is equal to the length of the encoded message, then there is no redundancy.

Lecture 37

1.
2. your algorithm may be able to encrypt and decrypt without a key for example – your algorithm may flip the bits. You do not need a key for that to decrypt. Just flip them back.
3. Hopefully just hides it and leaves the information intact.
4. If we know in English the number of e's is higher than we can assume that the highest number of symboled in the code are e's that is an example of how context clues can use redundancy as another clue for decoding.

Lecture 38

1. P=D(E(D(E(P))))
2. D(E(C,Ke),Kd)
3. To find clues about the scenario. In crises messages become more frequent so that might alert you to something.
4. if you know the language you also know things about it like, words, letters, frequency of each of those. Ect. That can all be used as clues for decrypting.

Lecture 39

1. A stream of bits, say 256 of them will take 2^256 possibilities to break. Although technically

breakable, it would take longer than a lifetime.
2. Linear Search requires you to go one by one so 2^n-1 times.
3. Substitution replaces symbols but that might be noticeable and breakable by itself since the symbols are int eh same order. So we need to rearrange them.
4. Confusion is about masking the actual symbols with other symbols while diffusion is about moving the symbols.
5. confusion could be better (relative term) since it can be any number of symbols that may or may not pertain to the original language.

Lecture 40

1. Mono-alphabetic changes uniformly. Uses the same letter for each symbol.
Alphabetical changes dependent on place in plaintext.
2 The key is a table of 1-1 mappings.
3.
4. Letter = letter + 2 so if A =1 then and B = 2 C = 3. then A(1) → C(3).
5. 25-26 depends how you look at it.
6. No you probably don't have to try all combinations to break it.

Lecture 41
1. For each letter there is 26 options. So 26^3 = 17,576.
2. X can still be any of 26 letter but now y=y so we only have 25 options once. We lose 26+1 = 27 options.
3. I think so since we seem to know AES's algorithm and the cipher text it produces yet it is still secure.

Lecture 42

1. It can be any possible plaintext for the key which means there is no way to reduce it by only knowing the algorithm and cypertext.
2. one-time pad being random ensures there are no context clues about the key's properties that can be used to reduce.
3. Well if it can only be used once you must send the new key to the receiver with each message. The only way to do this is through a secure channel. But if you have this channel you might as well use it to send the original info.

Lecture 43

1. Letter frequencies are maintained by transposition.

Lecture 44

1. One time pads are symmetric since it uses the same bit stream.
2. Distribution is about accessing vs management which preserves their safety.
3. No because you need K-1s (private key) to encrypt it.
4. Not exactly comparable. Depends on the use of this system and which characteristics fit better. Symmetric is not good for something like bit coins for example since there are many users.

Lecture 45

1. Most use block ciphers because Its integrity from tampering is secure and uses diffusion.
2. Integrity is the significance of malleability. It it's malleable then it can be damaged without notice.
3. Homomorphic encryption is malleable by design.

Lecture 46

1. Sub Bytes – uses it's value as an index in a look up table. It is then replaced by that item in the look up table.
RoundKey – xor'd with 128 key derived from the original key.
2. Shift rows – shifts each row by I bytes starting with 0 and going to Rn.
Mixedcolumns – replaces column by multiplying a 4x4 matrix of ints.
3. mixedColumns requires the inverse matrix which is not optimal for multiplication.
4. 128 bit blocks of info come in then various rounds (10-14) of 4 steps is applied to the bloc to systematically scramble the bits.
5. More rounds will further scramble the bits.

Lecture 47
1. identical blocks in the plaintest yield identical blocks in the ciphertext
2. randomize the blocks such as what CBC does by using the previous (or vector IV) to first scramble the bits systematically before encrypting.
3. You can potentially observe changes and content leak using XOR and multiple cipher blocks.
4. The cipher is used as a pseudo-random number generator that gives a one-time pad type of thing.

Lecture 48
1. Your private key that is used to decode the messages. The public key is used for encryption and is freely available to anyone.
2. one way functions allow people to compute the encryption and description easily so long as you have a key but makes it near impossible to without.
3. the problem was how to get the key to users securely. Well with a public key you can just send it whether it's secure or not.
4. can be 10000:1 for asymmetric vs symmetric. Aka public key takes a lot longer and is more complex than symmetric.

Lecture 49

1. Yes because the process is symmetrical to decode and encrypt.
2.
3. Theoretically no since it is based on unsolvable problems such as the knapsack problem.
4. Because they do not have A's key.
5. Not sure who from exactly because anyone can have A's public key.
6. A is sure it's B because no one else has B's private key.
7. Anyone with B's public key can decrypt the message.
8. could use two pairs of key's. One for privacy and the other for “signing”

Lecture 50

1. So that it is fast
2.
3. preimage says hard to find any m given an h where h= f(m) where are second preimage says it's hard

to find two different messages that have the same hash value.
4. 1.25*(2^64). aka really big.
5. 1.25*(sqrt(2^160)). Also really big.
7. preimage resistance allows us to ensure that M is bound to H(M)
8.

Lecture 51

1. Yes R has access to it's own private key and has access to S's public key.
2. Yes because it's RSA and R would still have access to all the necessary keys.
3. No one uses the senders private key.
4. Key exchange requires both confidentiality and authentication.

Lecture 52

1. They would not know a since a could be any number of integers.
2. still nothing as they would need to know g^b mod p. if they knew that then they could figure out the key.
3. if b were discovered they could use (g^a mod p) ^b to find the key.