CS361 Intro to Computer Security
Young, Bill

Daniel Rosenwald
June 26, 2014

EID: dpr447
CS login: randose
Email: danielrosenwald@gmail.com

Week 3

Lecture 34

1. It's impossible to transmit a signal over a channel at an average rate greater than C/h because that is the asymptote to which the average entropy approaches.
2. Increasing the redundancy can increase the reliability of a channel because a message is more likely to be received if it is said over and over again.

Lecture 35

1. h = -(log 1/10)
2. Computing the entropy of a natural language is difficult because there are so many factors that go into the probability of different letters appearing.
3. A zero-order model is the simple entropy by assuming all letters occur equally. First-order entropy takes into account the likelihood of certain letters following others, and second-order takes into account digrams that would block together, followed by trigrams in the third-order model.

Lecture 36

1. Prior probabilities can be impossible to compute because sometimes there is no prior knowledge with which to make those calculations.
2. The information content of a message is relative to the state of knowledge of an observer because the knowledge he/she has decides the probability of the occurrence of each message possible. This, in turn, affects the entropy.
3. If an encodings efficiency matches the entropy, there is no redundancy to compress out.

Lecture 37

1. The encrypted message has more than just letters – it also contains numbers and symbols like plus signs. The plus signs could indicate some kind of direction.
2. Encryption and decryption can be made to be self-sufficient algorithms that simply do the inverses of each other. Therefore, a key is not always necessary for these processes.
3. Encrypting a file obscures the information content of that file.
4. Redundancy gives clues because it allows the attacker to repeatedly see what happens in order and can be used as leverage to decipher what symbols are being put through and when.

Lecture 38

1. P
2. E(P, Ke)
3. Patterns can be used to detect reoccurring strings of a plaintext language from a ciphertext. A cryptanalyst would want to recognize patterns in encrypted messages to help decode them.
4. Properties of language, like grammar and sentence structure could help cryptanalysts validate or invalidate their theories on the algorithm they are trying to solve.

Lecture 39

1. Even if breakable, an encryption algorithm still may not be feasible to break because the breaker still won't recognize when he has reached success.
2. You'll find it on that order because one of them will produce the same results that you have in your plaintext. That order is just a brute force order, which isn't very good.
3. Substitution and transposition are both important in ciphers because they are the most basic building blocks of ciphers. Almost all commercial encryption algorithms use these two things in complicated ways to encode and decode data.
4. Confusion is the jumbling of information, while diffusion spreads out the information over a big ciphertext.
5. Confusion and diffusion are both important for encryption.

Lecture 40

1. The difference is that the former is just a mapping from one alphabet to another, while the latter treats each character separately based on where it is in the plaintext.
2. The key is the mapping of each letter of plaintext to ciphertext.
3. Because there are only so many letters to shift to.
4. The key is the shift distance.
5. [1,25].
6. No, not very strong because you'll pretty easily get the key.
7. Use the table in reverse.

Lecture 41

1. Because there are 26 possibilities for each of the 3 spots.
2. Because we know that the two y's correspond to the same English letter and a different one from the x.
3. Yes, Shannon proved it.

Lecture 42

1. Because even though you know the possibilities, you can't eliminate any of them for the key could be anything too.
2. The key must be random, otherwise you'd gain an advantage and could actually reduce the search space.

3. Sender and receiver must both know the key, but how do you securely get the key across? If there's already a secure channel, what do you need the key for, and if there isn't, how do you get the key there secretly?

Lecture 43

1. When using encryption by transposition, the letters move but they don't change. Therefore, you still get the same frequencies (i.e. lots of e's). That would give a decrypter the advantage of getting the language the plaintext is in.

Lecture 44

1. One-time pad is a symmetric algorithm because the key remains the same on both ends.
2. The former involves the problem of sharing the key between sender and receiver secretly, while the latter is the problem of making sure your large amount of keys are stored properly.
3. No, because it can only be decoded by S herself.
4. Neither are better – in fact, they're not easily comparable.

Lecture 45

1. I would assume most modern encryption algorithms are block ciphers because that allows transposition.
2. It allows an attacker to produce expected results on a plaintext from changing the ciphertext. This is a vulnerability to encryption algorithms.
3. Homomorphic encryption is the difference between being able to produce expected results from messing with a ciphertext and not.

Lecture 46

1. SubBytes uses confusion by substituting bytes for other bytes.
2. AddKey uses diffusion by spreading the data out.
3. Decryption is longer because the MixColumns inverse array has more difficult numbers to multiply.
4. Each block goes through a round of encryption, step by step.
5. You'd want to increase the number of rounds to jumble further the original message, making it harder for attackers to discover what the algorithm is doing to encode the plaintext.

Lecture 47

1. A disadvantage of ECB is that identical blocks in plaintext yield identical blocks in ciphertext.
2. This can be fixed by using CBC.
3. You can tell where change started, and if you have two identical ciphertext blocks you can find out information about plaintext blocks.
4. It is used as a PRNG in key stream generation.

Lecture 48

1. You must keep your private decryption key a secret.
2. They're critical because you need to ensure that your decoding can't be intercepted by an attacker.
3. Because it doesn't matter if someone else eavesdrops on your key – but only I have the private key. So no distribution is needed.
4. {P}K^-1
5. Asymmetric algorithms are much less efficient than symmetric algorithms, taking up to 10,000 times longer to encrypt and decrypt. However, they solve the painstaking work of key distribution and eliminate the problem of secure key sharing.

Lecture 49

1. Yes.
2. Prime numbers make finding the factors of large prime multiples extremely hard, which what RSA encryption relies on.
3. No, but it's easy to check.
4. Because you don't have the private key of the recipient.
5. Because we don't for sure know what B's private key.
6. Because we know the public key that it was encoded with – it was signed.
7. We would need the recipient's private key.
8. By using it's public key to sign and allowing it to only be decodable with A's private key.

Lecture 50

1. So you can easily go back and find the original.
2. Strong includes duplicates of the same plain message while weak only includes different ones.
3. One deals with multiple iterations of the hash while the other just deals with one hashing.
4. It means you'll hit a collision at approximately 1.25 * 2^64 items in.
5. 1.25 * 2^80
6. Because the hash function can be changed easily.
7. By re-hashing after every change made, we can see whether or not changes have been made such as tampers.
8. You could send a cryptographically hashed file through RSA and decode it on the other side in reverse order.

Lecture 51

1. No, because the attacker could just use S's public key and R's public key to decode the key.
2. Yes.
3. No, we'd actually have to use S's private key to decrypt the first one, which we don't have.
4. Both confidentiality and authentication – to ensure we cover the bases of mutual suspicion.

Lecture 52

1. The eavesdropper would still not figure out the value.
2. Then he could figure out the shared secret.
3. Similarly, he would figure out the shared secret.