Colin Murray
UTEID: cdm2697
UTCS-username: tashar
Email: murray.colin43@gmail.com

CS361 Questions: Week 3

The questions marked with a dagger (†) require external research and may be more extensive and time consuming. You don't have to do them for the assignment but, but do them to increase your competency in the class.

Lecture 34

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

C is the maximum transmission rate of the channel in bits per second, and h is the entropy (the most efficient encoding scheme for a language) in bits per symbol. It is clear that in order to end up with a rate faster than C/h you either have to reduce h or increase C. Since h is theoretically impossible to outperform (no encoding scheme can be more efficient than the calculated entropy) and C is simply the hard limit for transmission rate on a channel, it is not possible to increase C or decrease h and thus impossible to do any better than C/h.

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

Making an encoding scheme redundant means adding redundant repetition into a symbol's encoding such that even if most of the transmission is lost, the message can still be reliably decoded. It would be equivalent to retransmitting a single encoded symbol over and over until one transmission made it through my happy chance.

Lecture 35

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

$h = -(\log 1/10) \approx 3.219$

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

While we may only have 26 letters (27 with spaces) there is an enormous number of possible arrangements of these letters. While an individual letter may be relatively uncommon it might appear commonly in certain phrases. Dialectic trends may cause certain series of letters to occur more often than others. Context of the text also can have a heavy influence on which words occur more frequently. How to determine if redundant information can be reduced or eliminated without compromising the meaning of the message also plays an important role. There are simply a massive number of variables to account for.

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

Zero-order does not acknowledge frequency of symbols into the entropy calculation, leading to essentially a naïve encoding.
First order takes into account the relative frequency of each symbol and determines the entropy based on the relative frequencies.
Second order has a "memory" of the symbol that came before it and factors in a conditional probability. That is, if one symbol is transmitted, what is the likelihood that another symbol will follow it.
Third order extends the conditional probabilities introduced in Second order another step. Given one symbol, what is the probability that 2 particular symbols follow it.

Lecture 36

1. Why are prior probabilities sometimes impossible to compute?

The prior probabilities of some message or event can rely heavily on the extent of knowledge an observer has over the event and likewise the probabilities associated with the possible messages. Often the number of factors that influence these probabilities can be innumerable and at times such factors' influence can be unquantifiable.

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

The knowledge of an observer can influence how they perceive the relative frequencies of events. If relative frequencies are known, the information content becomes biased toward an event that is known to have a higher probability, and thus the information content is reduced to a smaller subset of possibilities (or at least a smaller subset of probable events). In an extreme case the message is already known and the information content becomes zero.

3. Explain the relationship between entropy and redundancy.

If a given encoded message exceeds the information content of a message (that is the amount of bits necessary to losslessly encode a message based on the entropy of the language) then there must be some redundancy in the encoding that may be eliminated while still preserving the meaning conveyed.

Lecture 37

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

If the encrypted message or its context can reveal any information about the possible plaintext, even very small details, it could be a useful tool at narrowing down the set of probable plaintexts for the encrypted message. Even a factor as small as the length of the plaintext or the language make is much easier to brute-force after all other hints are exhausted.

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

Encryption and decryption relies on the notion that the recipient shares enough information with the sender so the ciphertext can be decrypted and understood. While other methods may exist, like each party having a custom algorithm that nobody else knows about, a very simple and effective method is to have each party share a secret key, such that anyone without the key cannot decipher (or decrypt) the message.

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

The information content is preserved, but hidden from anyone that lacks the knowledge of how to decrypt it.

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

If redundancy in the plaintext is reflected in the ciphertext, hints about the nature of the plaintext can leak out. Assumptions of what likely plaintext produced the observed the redundancy seen in the ciphertext can be used to narrow the range of probable plaintexts and create a weak-point for an attacker.

Lecture 38

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

P

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

$E(P,K_E)$

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

Their goal is to break the encryption on a message to determine a weakness in the encryption method or how it is being used. Any sort of information related to the plaintext being somehow leaked in the ciphertext can potentially be used as leverage by an attacker to break the encryption or even recover the encryption key used. To design effective encryption methods the ciphertexts produced must undergo rigorous tests to hopefully prove that not a single bit of information related to the plaintext is leaked in the ciphertext.

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

Likelihood of certain words in a language, the probably ordering of certain words and the frequency of letters inside these words can reduce the possible range of plaintexts quite a bit, especially if any of these patterns are reflected in the ciphertext.

Lecture 39

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

Any encryption algorithm that exists today can be brute-forced to decrypt the ciphertext. If it is know that an encryption algorithm uses a 10-bit key then the attacker (who knows the algorithm, which is likely as most reputable crypto-algorithms are public) can try all 2^{10} possible keys until he finds one to decrypt it. This might be impossible for a human but a computer can do this very quickly. This is why many encryption algorithms today use very large keys like AES-256 (256 bit key). With 2^{256} possible keys one source suggests it would take 3.31×10^{56} years to crack the AES encryption key with a modern supercomputer (to put in perspective this is a much larger number than the current age of the universe in attoseconds (1 attosec = $1x10^{-18}$ seconds)).

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

In order to make sure an attempted decryption with a guessed key is successful, an attacker would need to know the plaintext corresponding to a sample cipher text. Once he can use this known plaintext he can attempt to decrypt the ciphertext continuously with random keys until the resulting decrypted text matches the known plaintext. If K is n bits long than there are 2^n possible K values the attacker must test, but in all probability he will find it at least half-way through his search. Thus on average the attacker must try 2^{n-1} different keys before he breaks the encryption.

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

Nearly all modern encryption algorithms today use some combination of substitution and transposition. Both are effective at scrambling the meaning of a given message and reversible without permanently losing some of the message's information.

4. Explain the difference between confusion and diffusion.

Confusion means replacing symbols in the plaintext with something else to confuse the attacker such that the information is not easily extractable. Diffusion takes the information content of a particular location in the plaintext and distributing it widely over the ciphertext so there is no tell-tale signature of this symbol or piece of information in any particular location to be discovered in the ciphertext.

5. Is confusion or diffusion better for encryption?

Both are important for an effective encryption algorithm. Confusion can hide the explicit meaning of the plaintext and diffusion can hide the semantics and patterns naturally occurring in the language used.

Lecture 40

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

A monoalphabetic cypher has a one-to-one replacement of symbols in the plaintext to create the ciphertext. Polyalphabetic ciphers replace symbols in the plaintext with different symbols depending on where the substation occurs.

2. What is the key in a simple substitution cipher?

Whatever method used to specify the 1-1 mapping of symbols in the plaintext and ciphertext (a table of mappings for instance).

3. Why are there k! mappings from plaintext to ciphertext alphabets in simple substitution?

Given an alphabet size k, once the first mapping is decided upon only k-1 letters remain that are eligible for the next mapping, and so on. This gives a total mappings of (k)(k-1)(k-2)…(1) or k!

4. What is the key in the Caesar Cipher example?

The number of positions in the alphabet you choose to shift your plaintext letters.

5. What is the size of the keyspace in the Caesar Cipher example?

The size of the alphabet being used.

6. Is the Caesar Cipher algorithm strong?

No, it's likely you wouldn't have to try every key in the keyspace in order to find the correct mappings.

7. What is the corresponding decryption algorithm to the Vigenere ciphertext example?

For each letter in the ciphertext you would find the corresponding letter in the key and use the pair of letters as row and column indices. The letter at the specified row and column is the corresponding letter of plaintext for the given ciphertext and key.

Lecture 41

1. Why are there 17576 possible decryptions for the "xyy" encoding on slide 3?

Using a substitution cipher (not a simple substitution cipher) there are 26 possibilities of letters 'x' had been substituted with, 26 for the first 'y' and 26 for the 2nd 'y'. This gives 26^3 possible descriptions of the encoding "xyy", or 17576.

2. Why is the search space for question 2 on slide 3 reduced by a factor of 27?

This supposes that the information was encrypted using a simple substitution cipher. Using this method there are 26 possible substitutions for 'x' and only 25 remaining possible substitutions for 'y' (both 'y' letters correspond the same letters in the plaintext). This gives 26 * 25 = 650 possible decryptions.

3. Do you think a perfect cipher is possible? Why or why not?

Yes, but these ciphers are very expensive and as such not as practical as other "good enough" ciphers. Take for instance a message encrypted using a truly random one-time-pad that is just as long as the message. True randomness is hard to come by, however, so such a cipher would be impractical for many modern-day crypto systems that must not sacrifice too much in the way of availability.

Lecture 42

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

Because the key used to encrypt the plaintext is truly random and just as long as the plaintext itself. Thus with a 15bit string there are 2^{15} conceivable plaintexts and likewise 2^{15} conceivable random keys. Knowing the algorithm does not aid in determining what key was used, and thuse doesn't help in deciphering the ciphertext. Likewise, in observing the ciphertext the attacker cannot narrow the range of possible plaintexts because each bit of the ciphertext depends just as much on the plaintext as it does on the random key. Since the random key cannot be narrowed down based on its representation in the ciphertext (a truly random key would have no patterns to make conclusions off of), neither can the plaintext.

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

If the key isn't truly random, the attacker may be able to get some sort of information that narrows the keyspace. This breaks perfect encryption which states that no information should be leaked about the plaintext (or the key) by examining the ciphertext.

3. Explain the key distribution problem.

Both the sender and the receiver must have this "one-time-key" otherwise the receiver cannot decrypt the ciphertext. Either a secure channel must be used to communicate the key, at which point why not just use the channel the communicate the message (which is just as long), or some other means must be used to secretly communicate the key (which as mentioned, is just as long as the plaintext so why not use the secure channel to communicate the plaintext).

Lecture 43

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

Transposition only shifts the ordering of characters around in the text, but does not replace any characters. This means the original frequencies of each symbol in the text is retained. This would give at least some information about the semantics of the plaintext (language used for instance). If the ciphertext seemed to have this property it would also clue the attacker in that a transposition was used to encrypt which aids the attacker in choosing an effective method of deciphering.

Lecture 44

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

Symmetric algorithm.

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

Key management deals with keeping track of which keys correspond to which senders so the proper key is used to decrypt their message. This is particularly important as most modern systems must keep track of a large amount of different keys. Key distribution is only concerned with a single key and how to communicate this key from sender to recipient so both parties can securely communicate.

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

No, every public key K_S has a corresponding private key K_S^{-1} that is computationally infeasible to derive just from K_S alone (with modern symmetric algorithms at least). Any message properly encrypted with K_S cannot be decrypted with any value other than K_S^{-1} which should only be known by the party that originally computed the K_S, K_S^{-1} pair to begin with.

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

Both serve very different purposes and work together very well in modern security systems like SSL. Symmetric encryption keys are very simple to generate whereas public keys require a good deal of computation go generate. This is compounded by the fact that the size of the keyspace for a Public Key system is much smaller for a set amount of bits than it is for a Symmetric encryption system (128-bit symmetric key may have equivalent keyspace to a 3000-bit public key). Public keys allow the number of key's managed per system to be greatly reduced (O(n)) whereas symmetric encryption systems require everyone to remember everyone else's key ($O(n^2)$). Public keys solve issues of key distribution and key management, though they are much more expensive than symmetric keys and are as such often used long-term to establish short-term symmetric encryption keys between parties.

Lecture 45

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

The performance gain of using stream ciphers over block ciphers and the reduced chance of error propagation are not as high of a priority with the speed and reliability of computers today. Adding parity, checksums and/or hashes can identify errors to transmission can help reduce error propagation risk and much faster processing can make the performance gains relatively minor. Block cipher's resilience to tampering and a better scrambling of the plaintext with high diffusion. The improved security I would suppose trumps the benefits of stream ciphers in most cases.

2. What is the significance of malleability?

If chances to the ciphertext make predictable chances in the plaintext (malleability) an attacker could strategically insert new ciphertext which might chance the meaning of the entire ciphertext

transmission. If a disgruntled employee found the encrypted payroll file and was able to decipher which line corresponded him his entry, it would be bad if he found out how to change say $25,000 to $250,000 by predictably altering the ciphertext in a way that adds an extra 0.

3. What is the significance of homomorphic encryption?
Homomorphic encryption schemes are malleable by design and ensures that similar algebraic operations to be performed on both the plaintext and ciphertext. RSA is a good example where modular exponentiation is applied on both the plaintext in order to encrypt and ciphertext in order to decrypt. Other uses include creating secure voting systems, collision-resistant hash functions and private information retrieval schemes.

Lecture 46

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

The first step.

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

The second step.

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

The InverseMixColumns step in decryption is more computationally expensive.

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

A block is arranged as 4 x 4 array of bytes (128 bits). This block undergoes several rounds (depending on key size used), each consisting of 4 steps. These steps are: subBytes, shiftRows, mixColumns and addRoundKey. Every round scrambles the information of the plaintext more and more in an invertible way while mixing it with round keys derived from the original key (in the addRoundKey step).

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

Every additional round in AES scrambles the information even more, making it harder to invert by brute force.

Lecture 47

1. What is a disadvantage in using ECB mode?

Identical blocks in the plaintext results in identical blocks in the ciphertext. This makes it easy to find redundant blocks and perhaps gather some information about the overall plaintext.

2. How can this flaw be fixed?

Each the encryption of each successive block relies some way of combining it (usually xor) with cipherblock behind it. A non-secret IV can be used on the very first block to kick off the process.

3. What are potential weaknesses of CBC?

If all the plaintext blocks are the same in a message the ciphertext will look the same (assuming the same IV and secret-key are used). This also means that if all the plaintext blocks are the same up until a point where a change has been made, the ciphertext will be identical up until the point where the plaintext differed.

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

Key stream generation uses a CBC like method to generate a random looking stream of bits block by block based on a key. This can be used as a one-time-pad with some plaintext message. In standard block encryption each block of plaintext is worked into the algorithm to produce the corresponding block in the ciphertext.

Lecture 48

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

The secret key (K_S^{-1})

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

For practicality, it should be easy enough to compute the encryption for a message in a public key system, but it should be computationally infeasible to retrieve the plaintext of what was encrypted without access to the private key. Additionally, public key cryptography schemes rely on other mathematical problems exist that are computationally easy to do, yet infeasible to invert (such as factoring large primes which could make it trivial to break RSA).

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

It is possible to encrypt messages that only someone with the correct private key can decrypt based on a publicly available public key. There is no need to secretly distribute keys which is a large problem with symmetric encryption schemes.

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.

The process of encryption and decryption in symmetric encryption is extremely fast, requiring only very quick arithmic operations that modern processors are very good at. Asymmetric

algorithms can take up to 10,000 times as long as the computations driving them are much more taxing on the processor (including modular exponentiation with very large numbers).

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, one can encrypt with the public key and decrypt with the private key OR encrypt with the private key and decrypt with the public key.

2. Explain the role of prime numbers in RSA.

Two distinct large prime numbers are chosen (p and q) such that n = pq. The public key (e) is chosen based on $\varphi(n) = \varphi(p)\varphi(q) = (p - 1)(q - 1) = n - (p + q - 1)$, where φ is Euler's totient function. Finally the private key (d) is chosen such that it is the multiplicative inverse of e (multiplicative inverse states that $d*e = 1 \bmod(\varphi(n))$). RSA is completely reliant on finding these 2 distinct large primes which are used to derive the public and private keys.

3. Is RSA breakable?

RSA relies on the unproven difficulty of factoring large numbers. The keyspace can still be brute forced so in that sense it is breakable. As of yet no one knows an efficient algorithm to factor large numbers and many very smart people have tried, thus we can assume RSA is secure.

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

Only a can decrypt anything encrypted with a's public key.

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

Presumably K_a (A's public key) is known by more people than B, so anyone can encrypt any message they want and send it to A. Encrypting with the public key offers no authentication of the sender from the recipient's perspective.

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

Only b knows his private key K_b^{-1}. Because of this, only B can encrypt something with his own private key that can be decrypted by anyone who knows B's public key. If the message can be decrypted using B's public key, A can be sure that the message came from B.

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

Anyone who has access to B's public key (kb) can simply decrypt the message like so: $\{\{M\}_{Kb^{-1}}\}_{Kb} = M$. Encrypting with the private key offers authentication to whoever reads the message, but no confidentiality since anyone can decrypt it.

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

Assuming A knows B's public key he can send a message like so: $\{B, \{M\}_{kb^{-1}}\}_{ka}$. A will decrypt the message using their private key: $\{\{B, \{M\}_{kb^{-1}}\}_{ka}\}_{ka^{-1}} = B, \{M\}_{kb^{-1}}$ and then look up the public key for B in order to decrypt the message like so: $\{\{M\}_{Kb^{-1}}\}_{Kb} = M$. Nobody can decrypt the original message but A (providing confidentiality) and nobody could have sent the message but B (providing authentication).

Lecture 50

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

Hash functions are intended to take in ANY arbitrary amount of data and compute some unique value that depends on what input was taken in. There are many applications of hashing data and performance is important so that including hashes doesn't significantly impact performance.

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

Weak collision resistance suggested given m1 it is hard to find m2 such that h(m1) = h(m2) and m1 != m2. In strong collision resistance you can choose m1, that is it is hard to find two messages m1 and m2 such that h(m1) != h(m2) where m1 != m2.

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

Preimage resistant suggests given a some h, it is hard to find any m that hashes to that h (h = f(m)). Second preimage resistance suggested given m1 it is hard to find m2 such that h(m1) = h(m2) and m1 != m2.

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

A collision can be found after $1.25\sqrt{2^{128}} = 1.25*2^{64}$ attempts.

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

A collision can be found after $1.25\sqrt{2^{160}} = 1.25*2^{80}$ attempts.

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

Cryptographic hash functions cannot be inverted. Identical messages will hash to the same value, but given a hash there is no way to pull out the original message.

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

Collision resistance assures tamper-resistance. Any alteration of M will create M^1. If the hash function has at least weak collision resistance it will be hard to alter M in such a way that the new M^1 results in the same hash as before (e.g. $H(M) = H(M^1)$)

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

$\{M, H(M)\}_{Ka}$
Once A receives this message they can decrypt it with their private key. After they can recomputed the hash of M and check that it matches the hash included in the message (giving integrity protection).

Lecture 51

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

No, anyone with access to S and R's public keys would be able to decrypt the message since the message is encrypted only with their private 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?

No, anyone then could use S's public key to strip off his "authentication" encryption and replace it with their own, that is to encrypt the message with their private key and send it as if it came from themselves.

3. Is $\{\{\{K\}_{KS^{-1}}\}_{KR}\}_{KS}$ equivalent to $\{\{K\}_{K^{-1}S}\}_{KR}$?

No, the first message cannot be decrypted by R since it requires S's private key in order to strip off the outermost encryption. Once that is done the messages would be equivalent, but as is they are not.

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

The key must be sent in a way that assures the authenticity of the sender and receiver, but also protects the confidentiality of the key being sent. RSA can achieve this by first encrypting the key with the sender's private key and then encrypting the result with the recipient's public key. The recipient is authenticated since he alone can decrypt it and the sender is authenticated because the key can only be decrypted with the sender's public key.

Lecture 52

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

Nothing, these are intended to be public.

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

If a is somehow discovered by an eavesdropped the eavesdropper can compute the shared secret by intercepting $g^b \bmod p$ and applying $(g^b \bmod p)^a \bmod p$. By doing this the eavesdropper now has the shared secret key between A and B.

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

If b is somehow discovered by an eavesdropped the eavesdropper can compute the shared secret by intercepting $g^a \bmod p$ and applying $(g^a \bmod p)^b \bmod p$. By doing this the eavesdropper now has the shared secret key between A and B.