MASTERNOTES

INTRO TO COMPUTER SECURITY

JUNG EUN YOON

[#1] LECTURE 1

- What is security?
  - "protection of assets against threats"
    - assets and threats are different for different contexts (i.e. data in computer security)
  - Security has many different contexts (i.e. personal, corporate, homeland, network)
- Attacks are more common now because of
  - Increased connectivity
  - More valuable assets online
  - Sophisticated tools invented
  - Low threshold to access
- Facts:
  - Over 1 million new malware
  - PCs stay infected (average 300 days)
  - 97% of sites vulnerable to malware
  - incorrect security code account for 80% of air force's vulnerability
- Protection in computer security:
  - **Privacy:** protecting anyone from seeing data (confidentiality)
  - **Integrity:** protecting anyone from changing data
  - **Availability:** being able to get to site
  - **Authentication:** protecting against phishing (stolen identity)
  - **Authorization:**
- Why learn security?
  - Enhance protection
  - Security in workplace, cyberspace
  - Quality/safety of transactions
- *Quotes:*
  - *"A dozen determined computer programmers can, if they find a vulnerability to exploit, threaten the United States' global logistics network, steal its operational plans, blind it's intelligence capabilities or hinder its ability to deliver weapons on target." – William J. Lynn, U.S. Deputy Secretary of Defense, Foreign Affairs*
  - *"A top FBI official warned today that many cyber-adversaries of the U.S. have the ability to access virtually any computer system, posing a risk that's so great it could 'challenge our country's very existence.'" – Computerworld, March 24, 2010*

→ QUESTIONS:

1. What uses of the term "security" are relevant to your everyday life?
→ As an individual, personal security is relevant. As someone who is constantly visiting web pages that ask me for my personal information, I expect to be secured by these sites (i.e. Amazon).

2. What do these have in common?
→ Other than that they all use the word 'security', it means different things in different contexts. However, what they have in common is the general definition: "protection of assets against threats."

3. Have you been a victim of lax security?
→ Yes, there was a time when the EE department accidentally sent out a mass e-mail with everyone student's names and e-mails. The dean only told us to "ignore" the e-mail and to delete it.

4. What is the likelihood that your laptop is infected? How did you decide?
→ Not too likely considering that it's a Mac and that it's pretty new. I used an anti-virus detection software to check right now.

5. What security measures do you employ on your laptop?
→ There's a built in firewall to Macs that I turned on, but nothing else otherwise.

6. Do you think they are probably effective?
→ Not particuarlly because there hasn't been a chance for it to prove its worth, but it probably helps that it is less likely for Macs to become infected with malware.

7. Consider the quote from the FBI official on slide 10. Do you think it overstates the case? Justify your answer.
→ It's true that most systems are vulnerable but it would be extremely difficult to take over an entire country, especially since there are people on the other end protecting the systems. It's not like we left the systems completely bare and open for everyone.

8. What is the importance in learning about computer security?
→ Learning about computer security enhances protection and security in the workplace and cyberspace. It also increases the quality and security of transactions.

[#1] LECTURE 2

- Why is security so hard?
  - **Perfect security is impossible**
  - Security is about ensuring that bad things never happen
  - You have to fix bugs and find identity features in the system that people could misuse/abuse
  - You have to identitfy all possible attacks before the attacker thinks of them whereas the attacker only needs to find one vulnerability
  - Systems are complex and involve data, people, software, hardware, media, etc.
    - It doesn't matter how much security you have on everything if you overlook one thing
  - Security is an afterthought – a system's main purpose isn't to be secure
- Tradeoff:
  - One side-effect of security is that it prevents important things from happening
  - Tradeoff between security and other goals (i.e. functionality, simplicity, usability, efficiency, production time, etc.)
- Quotes
  - "The three golden rules to ensure computer security are: do not own a computer; do not power it on; and do not use it." –Robert H. Morris, former Chief Scientist of the National Computer Security Center
  - "Unfortunately the only way to really protect [your computer] right now is to turn it off, disconnect it from the Internet, encase it in cement and bury it 100 feet below the ground." –Prof. Fred Chang, former director of research at NSA

→ QUESTIONS:

1. Consider the five reasons given why security is hard. Can you think of other factors?

→Systems are always expanding so security itself has to cover a larger field constantly. The job is never done.

2. Is there a systematic way to enumerate the "bad things" that might happen to a program? Why or why not?

→No, there is always another "bad thing" that might happen that no one has thought up of. There are an infinite number of "bad things".

3. Explain the asymmetry between the defender and attacker in security.

→The defender has to think of all the "bad things" that might happen in a system whereas the attacker need only find one vulernability.

4. Examine the quotes from Morris and Chang. Do you agree? Why or why not?

→Yes, perfect security is impossible. You can buy all the antivirus software out there but some attacker out there has thought of a way to intrude your computer. Their views are cynical but true.

5. Explain the statement on slide 8 that a tradeoff is typically required.

→Security has side-effects that prevents other important function goals from happening (usability, efficiency, simplicity, etc.) so a tradeoff is required. You are never going to have perfect security so you might as well forfeit some of It to invest in the other goals.

[#1] LECTURE 3

- perfect security is impossible so security is **risk management:** identifying and addressing risks in an enviroment
- **risk:** possibility that a particular threat will adversely impact an information system by exploiting a particular vulnerability
- **risk management framework (steps):**
  - access assets
  - access threats
  - access vulnerabilities (flaws)
  - access risks
  - prioritize countermeasure
  - make decisions
- **what do you do with risks?**
  - Acceptance: nothing you can do about it; countermeasure costs too much
  - Avoidance: not doing something that could cause risk
  - Mitigation: doing something to reduce chance of loss due to risk
  - Transfer: shift to risk to someone else (i.e. insurance, home security systems)
- **ALE (Annualized loss expectancy)**
  - (how to figure out risk for a particular scenario)
  - Gets expected value of loss
  - Per risk → Risk, whole loss if risk occurs, probability that it might occur (how many times per year), and cost per year
  - Downsides:
    - Naïve
    - Only looks at expected value
  - You have to access other things besides risk and espected value (i.e. Technical, economic, psychological)

→ QUESTIONS:

1. Define "risk"?

→ The possibility that a particular threat will adversely impact an information system by exploiting a particular vulnerability.

2. Do you agree that software security is about managing risk?

→Yes, perfect security is impossible and that leads to tradeoffs, which is another way of saying that software security is about managing risks.

3. Name and explain a risk you accept, one you avoid, one you mitigate, and one you transfer?

→A risk that I accept is driving, though there's always the possibility of getting into an accident. Risk avoidance: I don't drink when I'm drunk and I don't speed. Risk mitigation: I chose to drive an old, rundown car instead of a nice new car to avoid money loss. Risk transfer: I use State Farm insurance.

4. Evaluate annualized loss expectancy as a risk management tool.

→It is a table of possible losses, their likelihood, and potential cost for an average year. It's a pretty good idea but also fairly a naïve idea. The probability makes it seem like you should only invest your time only on the one with the highest ALE but that's not true at all. Just because one risk has a low probability doesn't mean that you shouldn't put some time into it.

5. List some factors relevant to rational risk assessment.

→Technical, economic, psychological factors are relevant to rational risk assemesment.

[#1] LECTURE 4

- Aspects of security:
  - **Confidentiality** (who can read info?)
    - First computer security problem
    - For military
    - Not all data is equally sensitive
    - How to authorize who can see what?
    - Does authorization change over time?
  - **Integrity** (who can write/modify info?)
    - For commercial applications
    - How to detect unauthorized changes?
    - Who is authorized to modify?
    - Does authorization change over time?
  - **Availability** (are resouces there when I need them?)
    - DoS – denial of service attacks
    - Resources provided in timely fashion and allocated fairly?
    - Is system usable (UI/UX)?
    - Can system recover/compensate if problem occurs?
    - How is concurrency controlled by the system?
  - **Authentication** (who can establish identity)
  - **Non-repudiation** (who can deny my actions?)
- There is no most important aspect because it depends on the context
- **Mechanisms:** protects one or more of the major aspects
  - Crypotography
  - Digital signatures
  - Firewalls
  - Passwords
  - Certificates
  - Access control (login)

→ QUESTIONS:

1. Explain the key distinction between the lists on slides 2 and 3.
→One is a list of aspects of security and the other is a list of mechanisms, the aspects that protects one or more of the major aspects

2. Consider your use of computing in your personal life. Which is most important: confidentiality, integrity, availability? Justify your answer.
→I find that I need availability the most. Most of my computing life has to do with doing school work and I constantly need to know that I have all the resouces I need at a minute's notice.

3. What does it mean "to group and categorize data"?
→You want to separate your data to characterize which data is more sensitive and which ones are less sensistive for confidentiality and integrity.

4. Why might authorizations change over time?
→Data's sensitiveness might change over time so the system has to change with it. A person's rights/power might also change over time.

5. Some of the availability questions seem to relate more to reliability than to security. How are the two related?
→Reliability measures availability. A system can not be considered reliable when it is not available, and that is why it is so important in commercial cases.

6. In what contexts would authentication and non-repudiation be considered important?
→Authentication and non-repudiation is important to the registrar system itself, though they serve to proect confidentiality and integrity.

[#2] LECTURE 5

- **Security policy:**
  - how to define security for a given system since ***security itself is a very general notion***
  - set of rules for implementing specific security goals
  - a contract between the coder and the customer
  - realistic and efficient
- **Metapolicy:** overall, specific security goals; very general
  - if satisfied, then policies are sufficient
  - if not satisfied, then policies aren't doing enough and should be adjusted
- **Policy:** rules to achieve metapolicy (specific security goals)

→ QUESTIONS:

1. Describe a possible metapolicy for a cell phone network? A military database?
→ For a cell phone network, a metapolicy might be to protect cell phone numbers. For a military database, a metapolicy might be to protect the assignement locations of soldiers.

2. Why do you need a policy if you have a metapolicy?
→ The metapolicy is too general, leaving too much room for interpretations, and seems arbritary if you don't have policies.

3. Give three possible rules within a policy concerning students' academic records.
→ Only the registrar's personnel can change records; students and professors can not change records. Only students, the registrar, and authorized staff (i.e. advisors) can view their records. Students' must login view records.

4. Could stakeholders' interest conflict in a policy? Give an example.
→ Yes, students and their academic records. You can't block everyone besides the registrar's office from seeing the records because the stakeholders (the students) should be able to see their records, though they shouldn't be able to change them.

5. For the example given involving student SSNs, state the likely metapolicy.
→ Protecting students against identity theft.

6. Explain the statement: "If you don't understand the metapolicy, it becomes difficult to justify and evaluate the policy."
→ The matapolicy is very vague and it's left up to interpretation so you must understand it fully to jusitify it via policies.

[#2] LECTURE 6

- **multi-level security (MLS) / military security:** how do you control levels of access?
- **Labels:** has two parts → heirarchial linearly ordered set that tells us order of importance, and need-to-know categories
- **Mixed information:** (part of multiple labels) must be labeled to protect the information at the highest hierarchical level and protect all categories of information
    - if object is both high and low, make it high so that it's more protected than necessary rather than less protected

→ QUESTIONS:

1. Why is military security mainly about confidentiality? Are there also aspects of integrity and availability?
→ Military security is all about secrets and plans and how others could get to that secret information. We care about integrity and availability but it's mostly because confidentiality.

2. Describe the major threat in our MLS thought experiment. (Military Setting)
→ The confidentiality of information—no person not authorized to view a piece of information may have access to it.

3. Why do you think the proviso is there?
→ There will be some counterintutitive results if we include the other security aspects.

4. Explain the form of the labels we're using.
→ The label on any folder reflects the sensitivity of the information contained within that folder. The label contains both a hierarchical component and a set of categories. You can look at a label and see how sensitive it is.

5. Why do you suppose we're not concerned with how the labels get there?
→ The labels were put there by a random person. It could've been a security officer who has access to all the files. We only care about the labels themselves.

6. Rank the facts listed on slide 6 by sensitivity.
→ 1) The cafeteria is serving chopped beef on toast today, 2) The base softball team has a game tomorrow at 3pm, 3) Col. Smith didn't get a raise, 4) Col. Jones just got a raise, 5) The British have broken the German Enigma codes, 6) The Normandy invasion is scheduled for June 6.

7. Invent labels for documents containing each of those facts.
→ Unclassified (1,2), Confidential (3,4), Secret, Top Secret (5), Need-to-know (6).

8. Justify the rules for "mixed" documents.
→ You use the highest appropriate level, otherwise the documents wouldn't be protected as much as you need, and people who shouldn't be authorized to see the documents would see it. Better to be safe than sorry.

[#2] LECTURE 7

- People get **clearances/authorization levels** just like the folders; labels
- **Principe of least privilege:** any subject should have access to the minimum amount of information needed to do its job.
    - You can look down but not up

→ QUESTIONS:

1. Document labels are stamped on the outside. How are "labels" affixed to humans?

→ We give them clearances or authorization levels, of the same form of document sensitivity levels. These "lables" indicate classes of information that person is authorized to access.

2. Explain the difference in semantics of labels for documents and labels for humans.

→ When you label a document, you label the type of document within. But when you put a "label" on a human, it indicates the amount in which we trust that individual.

3. In the context of computers what do you think are the analogues of documents? Of humans?

→ Data would be the documents and users would be the humans.

4. Explain why the Principle of Least Privilege makes sense.

→ You should only be allowed to look down. An individual should never be allowed to look above his clearance level. That is, a lietenant shouldn't be able to look at the same files that the captain can look at.

5. For each of the pairs of labels on slide 6, explain why the answers in the third column do or do not make sense.

→ For the first and third pairs, the clearance level is above that of the sensitivity level so it's fine that the individual looks at those files. For the second pair, the sensitivity level is above that of the clearance level so it's uncessary for the individual to look at the file because any subject should have access to the minimum amount of information needed to do its job.

[#2] LECTURE 8

- Definitions:
  - **Objects:** information containers protected by system (documents, folders, files, etc.)
  - **Subjects:** entities (users, processes, etc.) that execute activities and request access to objects
  - **Actions:** operations, primitive or complex, executed on behalf of subjects that may affect objects
- **Dominates relation:**

  Definition: (L_1, S_1) *dominates* (L_2, S_2) iff
  1. $L_1 \geq L_2$ in the ordering on levels, and
  2. $S_2 \subseteq S_1$.

  We usually write $(L_1, S_1) \geq (L_2, S_2)$.
  - formalizes a relationship between any two labels
  - same thing as simple security
  - **superset:** a is contained in b
  - **Reflexive:** each level dominates itself
  - **Transitive:** x > y and y > z, then x > z
    - If there is a path using two or more arrows in the lattice, then there can be another arrow connecting that path
- **Simple security property:**
  - shows how to use dominates to decide whether a read access should be allowed

→ QUESTIONS:

1. Why do you think we introduced the vocabulary terms: objects, subjects, actions?
→ So far we have only talked about a specific case study but we need to be more general to describe security in various situtations.

2. Prove that dominates is a partial order (reflexive, transitive, antisymmetric).
→ Partial order means that there are labels where neither of them dominate each other. For example, top secret cryto and top secret nuclear do not dominate each other.

3. Show that dominates is not a total order.
→ If it isn't a total order, it is a partial order. Partial order means that there are labels where neither of them dominate each other. For example, top secret cryto and top secret nuclear do not dominate each other.

4. What would have to be true for two labels to dominate each other?
→ The labels have to be identical.

5. State informally what the Simple Security property says.
→ An individual who has a clearance level that dominates the sensitivity level of the document can read it.

6. Explain why it's "only if" and not "if and only if."
→ "if and only if" means that the condition is both necessary and sufficient. It is necessary that the subject level dominates the object level, but not sufficient. There may be other constraints in the system that prevent the read from happening, even if SS would allow it.

[#2] LECTURE 9

- **<u>*-Property:</u>**
  - restricts write access

    The *-Property: *Subject S with clearance* (L_S, C_S) *may be granted write access to object O with classification* (L_O, C_O) *only if* $(L_S, C_S) \leq (L_O, C_O)$.
  -
  - you can only write above or at your level

→ QUESTIONS:

1. Why isn't Simple Security enough to ensure confidentiality?
→Simple security only codifies restrictions on read access and you need to restrict write access as well.

2. Why do we need constraints on write access?
→Information can flow in both directions and read access only takes care of one of the directions.

3. What is it about computers, as opposed to human beings, that makes that particularly important?
→Subjects in the world are computing are often operating on behalf of a trusted user (and with his or her clearance). We not only have to worry about the people but also the programs on the computers that are running on their behalf.

4. State informally what the *-Property says.
→You can write above or at your level. It uses dominates to decide whether a write access should be allowed.

5. What must be true for a subject to have both read and write access to an object?
→The labels must be identical.

6. How could we deal with the problem that the General (top secret) can't send orders to the private (Unclassified)?
→The general can log out of his top secret account and login to an unclassified account and send the orders.

7. Isn't it a problem that a corporal can overwrite the war plan? Suggest how we might deal with that.
→We have to put others in place to deal with integrity because all of our rules are for confidentiality. We could have a policy that checks that it was the corporal who wrote it and not someone else via security questions but again, that would be a policy to enforce integrity and we're not interested in that yet.

[#2] LECTURE 10

- *-property and simple security doesn't take care of labels changing but **tranquility properties** do.
- **Strong tranquility:** subjects/object don't change labels during lifetime of system
    - once you define a label you don't change anything
- **Weak tranquility:** subjects/objects don't change labels in a way that violates the "spirit" of the security policy
    - Don't do something that you wouldn't want to happen in accordance to *-property and simple security
- **Bell and lapaula policy (BLP):** combines simple security, *-property, and tranquility properties
    - Constrains the flow of information among the different security levels within the lattice

→ QUESTIONS:

1. Evaluate changing a subject's level (up or down) in light of weak tranquility.
→Raising the subject's level is bad because then a lower level subject could view a higher level subject just by raising a level. Lowering the subject's level is tricky because it might be a write down if there's residual subject but a stateless subject with no residual subject would be fine.

2. Why not just use strong tranquility all the time?
→This is very restrictive. What if a user needs to operate at different levels during the course of the day?

3. Explain why lowering the level of an object may be dangerous.
→The information that is contained in a higher object would be accessible in a lower level, which would violate the security property.

4. Explain what conditions must hold for a downgrade (lowering object level) to be secure.
→You have to make sure that the security property still holds and the *-property still holds.

[#3] LECTURE 11

- **Access control policy**:
  - Introduces rules that control what accesses subjects may take with respect to objects
  - Ex: Bell and lapadula model
- **Mandatory access control (MAC):** rules are enforced on every attempted access, not at the discretion of any system user
  - What that means for BLP is that no access is ever allowed unles s it satisfies the Simple Security Property and *-Property.
- **Discretionary Access Controls (DAC):** rule enforcement may be waived or modified by some users

→ QUESTIONS:

1. Suppose you wanted to build a (library) system in which all subjects had read access to all files, but write access to none of them. What levels could you give to subjects and objects?

→Subjects could be H{} and objects could be L{nonempty} where L can be any level that is dominated by L. By the simple security property and *-Property, all subjects would have read access but not write access.

2. Why wouldn't you usually build an access control matrix for a BLP system?

→You may have thousands of subjects and thousands of objects so you would have millions of cells. Most of these cells would be empty so it would be a waste of time.

[#3] LECTURE 12

- **Lattice:** collection of security goals. We can leave off the reflexive and transitive arrows.
  - **Reflexive:** each level dominates itself
  - **Transitive:** x > y and y > z, then x > z
    - If there is a path using two or more arrows in the lattice, then there can be another arrow connecting that path
- **Path from L1 to L2:**
  - L2 dominates L1 so L1 does all the work
  - ***Metapolicy of BLP Lattice:*** Information should flow up, now down
  - READ: L2 pulls data from L1
  - WRITE: L1 pushes data to L2

→ QUESTIONS:

1. Suppose you had hierarchical levels L, H with L < H, but only had one category A. Draw the lattice. (Use your keyboard and editor to draw it; it doesn't have to be fancy.)

→L{}
H{}
H{A}
L{A}

Ab b

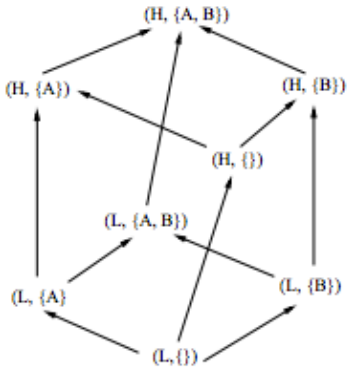
2. Given any two labels in a BLP system, what is the algorithm for finding their LUB and GLB?

→For any subset of the set of two labels you can find the least upper bound and a greatest upperbound. For the least upper bound. The join of two security levels is the LUB and the meet of two security levels is the GLB.

3. Explain why upward flow in the lattice really is the metapolicy for BLP.

→A path from L1 to L2 means that information is allowed to flow from L1 up to L2. This either means hat a subject at level L2 can read a level L2 object or a subject at level L1 can write a level L2 object, which demonstrates simple security and the *-Property.

4. Draw the lattice with level L,H with L < H and categories {A,B}. You can omit the reflexive and transitive arrows.



[#3] LECTURE 13

- **Operations:**
  - **READ(S,0):** if object O exists and Ls >= Lo, then return current value, otherwise, return a zero
  - **WRITE(S,O, V):** if object O exists and Ls <= Lo, change its value to V, otherwise, do nothing
  - **CREATE(S,O):** if object O doesn't exist, create new object O at level Ls, otherwise, do nothing
  - **DESTROY(S,O):** if object O exists and Ls <= Lo, destroy it, otherwise, do nothing.
- **Covert Channel:** If you have a low level and a high level subject, and the high level subject can do stuff and the low level subject can see that, then you can use that to send a bit of information from the high to the low

→ QUESTIONS:

1. Explain how the BLP rules are supposed to enforce the metapolicy in the example on slide 1.
→Given the example, the BLP rules should make it so that the flow only goes from L to H according do simple security. Thus, BLP does constrain the flow as it's supposed to.

2. Argue that the READ and WRITE operations given satisfy BLP.
→The READ and WRITE operations are just instances of simple security and the *-Property, and thus restrict the flow accurately.

3. Argue that the CREATE and DESTROY operations given satisfy BLP.
→The CREATE operation does not violate the BLP laws in any way. It is simply creating a new object should it not exist already in the system. It's not giving away anything. As for the DESTROY operation, it abides by the *-Property, being that it could technically be argued that DESTROY is a form of WRITE, so it also satisfies BLP.

4. What has to be true for the covert channel on slide 5 to work?
→It must still abide by the BLP rules and go by undetected in that sense.

5. Why is the DESTROY statement there?
→It is a covert channel and you don't want to leave a trace so you want to delete your tracks. The purpose of the new objects was to pass the bit of information so it has no further use after the bit has been transmitted.

6. Are the contents of any files different in the two paths?
→No. F0 is always the same; it's always just a value and then written as a 1. It's just how we interpret that information that gets us the differing transmitted bits.

7. Why does SL do the same thing in both cases? Must it?
→Because we inherently have to transmit to SL from SH, it does the same things and it must. SL must

8. Why does SH do different things? Must it?
→You can't write down so you can't literally transmit a 1 or a 0 from SH to SL so we have to use a workaround by using the BLP rules to our advantage. So, yes, SH must do different things.

9. Justify the statement on slide 7 that begins: "If SL ever sees..."
→If SL's results can be manipulated by SH, that is, if SL ever sees varying results depending on varying actions by SH, then we can use it to transmit bits of information between the two subjects. Though thi would violate the metapolicy, it wouldn't be against the BLP rules, making the operation convert/"undercover". That is to say, there may be other system features that could be manipulated to convey information.

[#3] LECTURE 14

- **Covert channel:** path for the illegal flow of information between subjects within a system, using system resources that wer not designed to be used for inter-subject communication
    - Illegal flow is beween subjects in system, not between humans
    - Flow violates metapolicy but might abide by policies
    - Flows in system resources that weren't designed as communications channels (i.e. flags, clocks, etc.)
- **Types of covert channels:**
    - **Storage**: subject stores data into the state of the system (status, etc.)
        - sender and receiver has access to shared object
        - sender has to be able to modify object
        - receiver has to detect change
        - mechanism to time accesses (coordinate data passing)
    - **Timing:** Information is recorded in the ordering or duration of events in system
        - sender and receiver has access to shared object and time reference (timer, clock, etc.)
        - sender has to control timing
        - mechanism to time accesses (coordinate data passing)
    - **Implicit:** ex
    - **Termination:** ex
    - **Probability:** ex
    - **Resource Exhaustion:** ex
    - **Power:** ex

→ QUESTIONS:

1. Explain why "two human users talking over coffee is not a covert channel."

→ The flow in a covert channel is between subjects within the system, not humans.

2. Is the following a covert channel? Why or why not?

Send 0 | Send 1

Write (SH, F0, 0) | Write (SH, F0, 1)

Read (SL, F0) | Read (SL, F0)

→No. Both would return a 0 in this case since you can only read down and if the second program returned a 1, it would violate the BLP rules and not only the matapolicy.

3. Where does the bit of information transmitted "reside" in Covert Channel #1?

→It is in the state of the system – the status of the resource.

4. In Covert Channel #2?

→ The information is recorded in the ordering or duration of events on the system. That is, q reads the bit by consulting the system clock to see how much ti me has elapsed since it was last scheduled.

5. In Covert Channel #3?

→Disk drive – there are aspects of both time and storage channel.

6. In Covert Channel #4?

→ In the control flow of a program – implicit channel.

7. Why might a termination channel have low bandwidth?

→Terminations might have low bandwidth because these channels need time exponential in the size of the leaked information.

8. What would have to be true to implement a power channel?

→A countable amount of energy would have to be consumed by the system.

9. For what sort of devices might power channels arise?

→Really, any system with power…which would be all systems. In particular, a block box might be more suspectable to power channels.

[#3] LECTURE 15

- **Impossible to remove all covert channels**
- **Cover channels are fast**
    - Operate at thousands of bits per second with no impact on system processing
- **What we care about:**
    - **Existence:** is the channel there?
    - **Bandwidth:** how much information can be sent over the channel?
    - **Noiseless/noisy:** if a channel is noisy, it's difficult to extract information
- Dealing with cover channels:
-

→ QUESTIONS:

1. Explain why covert channels, while appearing to have such a low bandwidth, can potentially be very serious threats.

→ It might seem that such channels would be so slow that you wouldn't really care but that's not true. Covert channels on real processors operate at thousands of bits per second, with no appreciable impact on system processing.

2. Why would it be infeasible to eliminate every potential covert channel?

→There are many factors but for one thing, the detection of a covert channel can be made extremely difficult by using characteristics of the communications medium that are never controlled or examined by the subjects/users (working against instrusion detection). Channels of these type can remained undetected for long periods.

3. If detected, how could one respond appropriately to a covert channel?

→ Dealing with a covert channel may include: eliminating it, restricting the bandwidth, or monitoring it. We can eliminate it by modifying the system implementation, reduce the bandwidth by introducing noise into the channel, or we can can monitor it for patterns of usage that indicate someone is trying to exploit it. This is intrusion detection.

4. Describe a scenario in which a covert storage channel exists.

→Suppose you have a high level and a low level subject and information isn't supposed to flow from high to low. Yet, the low level subject can try to access the high level subject and gets back one of two error messages: Resource not found or Access denied. If the high level subject can manipulate the bit of information sent in the resouces's status on each access attempt by the lower level subject, then this would be a scenario in which a covert channel exists.

5. Describe how this covert storage channel can be utilized by the sender and receiver.

→ Both sender and receiver must have access to some attribute of a shared object, the sender must be able to modify the attribute, the receiver must be able to reference (view) that attribute, and a mechanism for initiating both processes, and sequencing their accesses to the shared resource, must exist.

[#3] LECTURE 16

- **SRMM (Shared Resource Matrix Methodology):** matrix describes potential effects on shared attributes/objects (file size, level, existence, etc.)
    - for storage channels
        - R = References = basically READ
        - M = Modifies = basically WRITE
    - → **INTERPRET:** if R and M in same row, there is a potential channel. It's not concrete proof that the channel is there.
    - **drawbacks:** difficult to use correctly and effectiviely because you need to know a lot about the sematics of the system operations
        - you need a new matrix for each different system

→ QUESTIONS:

1. Why wouldn't the "create" operation have an R in the SRMM for the "file existence" attribute?
→If you do a create you should hae in R in file existence because it either already existed in which case the operation failed or it didn't already exist in which case we created it.

2. Why does an R and M in the same row of an SRMM table indicate a potential channel?
→Because the R and an M in a particular row, there is a mechanism in which someone can modify it and someone can reference it and that's what you need for a cover channel to exist.

3. If an R and M are in the same column of an SRMM table, does this also indicate a potential covert channel? Why or why not?
→No, each row has to do with different potential effects on shared attributes of objects. Rows don't have much to do with each other and should be thought of separately.

4. Why would anyone want to go through the trouble to create an SRMM table?
→ Kemmerer's Shared Resource Matrix Methodology provides a systematic way to investigate potential covert channels, so it ends up being a pretty nice tool, though to use it effectively it requires a lot of knowledge abou t the semantics and implementation of system operations. It's just a tool to find potential covert channels.

[#4] LECTURE 17

- **Non-Interference policy:**
  - Type of **information flow policies**
  - It's not about who can read a file, who can write a file
  - What we care about: where information can flow in the system
  - H is "interfereing with" L if H does something that L can see the results of
  - **Policy:** system specificies exactly which subjects can interefere with which subjects → where information can flow
    - Reflexive (S1 to S1)
  - You can take any MLS policy and turn it into a Non-Interference policy

→ QUESTIONS:

1. If a computer system complies with the BLP model, does it necessarily comply with non-interference? Why or why not?
→No, the BLP system allows for covert channels but the NI policy doesn't.

2. What would the NI policy be for a BLP system with subjects: A at (Secret: Crypto), B at (Secret: Nuclear)?
→A can communicate to B and vice versa (reflexive). That is, A → B and B → A.

3. Can covert channels exist in an NI policy? Why or why not?
→No, the NI policy is very strict and its purpose is to restrict interferences between subjects, meaning that a covert channel can not exist.

4. If the NI policy is A– > B, in a BLP system what combinations of the levels "high" and "low" could A and B have?
→Answer

[#4] LECTURE 18

- Fact
- Fact
- Fact

→ QUESTIONS:

1. Why do NI policies better resemble metapolicies than policies?
→Answer

2. What would be L's view of the following actions: h1, l1, h2, h3, . . . , hj, l2, l3, . . . , lk
→Answer

3. What is difficult about proving NI for realistic systems?
→Answer

[#5] LECTURE 19

- **Integrity questions:**
  - → about credibility → who can you trust?
  - Who can modify/supply data
  - How do you protect assets?
  - How do you detect unauthorized changes?
- **Confidentiality vs. integrity:**
  - Confidentiality --> information flowing where it shouldn't so you really need two parties (receiver and sender)
  - But with integrity, you only need one party; it can be breached from the system itself (sloppy program that overwrites all files)
- **Integrity principles:**
  - **Separation of duty** is having several different subjects must be involved to complete a critical function (i.e. two bankers need to sign a check before it's valid).
  - **Separation of function** is forbidding a single subject cannot complete complementary roles within a critical process. That is, separation of function is like checks and balances; it can't perform more than one important function.
  - **Auditing:** recoverability and accountability require maintaining an audit trail.

→ QUESTIONS:

1. Explain the importance of integrity in various contexts.
→In a real life setting, you want to know which newspaper to go for credibile news. In a commercial setting, you want to make sure that your product is free of human errors (i.e. dividing up tasks in the programming world).

2. Why would a company or individual opt to purchase commercial software rather than download a similar, freely available version?
→Because of the commercial software's integrity. That is, the commercial software is more trustworthy because of its reputation and value. Something that has market value must be trustworthy.

3. Explain the difference between separation of duty and separation of function.
→ Separation of duty is more like splitting up a function among subjects whereas separation of function is like not allowing a single subject to perform more than one function.

4. What is the importance of auditing in integrity contexts?
→ If something bad does happen then you can roll back and assign responsibility / take care of whatever the problem was.

5. What are the underlying ideas that raise the integrity concerns of Lipner?
→People inherently make mistakes and that could potentially be a huge integrity problem. You have to protect the system against that.

6. Name a common scenario where integrity would be more important than confidentiality.
→Generally in the commercial enviorment, perhaps at Apple when they are building a new iOS system.

[#5] LECTURE 20

- **Integrity labels have nothing to do with clearance levels!!!! (i.e. high trust but low sensititivity)**
- **Integrity labels:**
  - **Objects:** degree of trusthworthiness of information (i.e. gossip is under news reports)
  - **Subjects:** trustworthiness of its ability to produce/modify/handle data
- **Structure:**
  - Heirarchiacal component – level of confidence
  - Set of categories – things that are in level
  - i.e. expert: {physics} → expert trustworthiness in physics so maybe a physics professor
- **dominates relation:** exactly the same as BLP
- metapolicy: Don't allow bad information to "taint" good information. An alternative formulation is: don't allow in formation to "flow up" in integrity.
  - Low information to taint high information
  - **We don't want info to flow up**

→ QUESTIONS:

1. Give examples of information that is highly reliable with little sensitivity and information that is not so highly reliable but with greater sensitivity.
→ High reliability with little sensitivity: The President said that he wants us to have a nice day. Low reliability with high sensitivity: A rumor is going around that Al Qaeda is still alive and that ISIS is just a ruse.

2. Explain the dominates relationships for each row in the table on slide 4.
→ First row: Expert > Student and Physics is a superset of Physics so L1 dominates L2. Second row: Novice < Expert so L1 does not dominate L2. Third row: Student > Novice and the art is a superset of empty set so L1 dominates L2.

3. Construct the NI policy for the integrity metapolicy.
→ Information can only flow up. That is, low data shouldn't be able to taint high data. High subjects can't read down and they can't write up.

4. What does it mean that confidentiality and integrity are "orthogonal issues?"
→ It means that they have nothing to do with eachother. That is, their confidentiality label and their integrity label should be separate from each other.

[#5] LECTURE 21

- Ken **Biba** proposed three different **integrity access control policies**
  - Low water mark integrity policy
  - Ring policy
  - Strict integrity
  - **Difference:** amount of trust in subjects
- **Strict integrity:** (biba integrity / biba model)
  - **Metapolicy: DIRECTION FLOW IS DOWWNNNN**
  - Little trust in subjects because flow is constrained to move down
  - **Simple integrity property:**
    - READ(S,0) if i(s) <= i(o)
    - Read up
  - **Integrity *-property:**
    - WRITE(S,O,V) if i(o) <= i(s)
    - Write down
- **Low watermark policy:**
  - **Watermark policy:** labels can change in a system
    - **High water mark:** float up and stick at highest level
    - **Low water mark:** float down and stick at lowest level
    - **Can be reset**
  - **Low watermark policy:**
    - WRITE(S,O,V) if i(o) <= i(s)
    - If READ(S,O) then I'(s) = min(i(s),i(o) = subject's new integrity level after read
- **Ring property:**
  - **Read:** you can read any object, regardless of integrity levels
  - WRITE(S,O,V) if i(o) <= i(s)
- **A system with both BLP and strict integrity:**
  - Access only allowed when both BLP and Biba rules are met

→ QUESTIONS:

1. Why is Biba Integrity called the "dual" of the BLP model?

→The biba integrity policy is just the BLP with a change in direction of the arrows; it's exactly the "dual" of the BLP model.

2. Why in the ACM on slide 5 is the entry for Subj3 - Obj3 empty?

→Though they have the same integrity level, neither of the sets is a superset of the other so neither R nor W can take place.

3. If a subject satisfies confidentiality requirements but fails integrity requirements of an object, can the subject access the object?

→No, both rules need to be met to gain access, though the confidentiality labels and the integrity labels are separate entities.

[#5] LECTURE 22

- In above lecture notes

→ QUESTIONS:

1. What is the assumption about subjects in Biba's low water mark policy?
→The subjects are not trustworthy and can be corrupted "easily", just by reading corrupted information.

2. Are the subjects considered trustworthy?
→ He is not giving much credit to the subject. He's saying that if a subject accidentally reads a corrupted information then that subject would also be corrupted and that subject's integrity level should move down.

3. Does the Ring policy make some assumption about the subject that the LWM policy does not?
→The subjects have enough common sense to filter out bad information.

4. Are the subjects considered trustworthy?
→In this case, yes. They are able to read from any object so we are assuming that these subjects are trustworthy and that they know how to filter out bad information.

[#6] LECTURE 23

- **Lipner**
  - He had concerns for commercial data processing
  - How to put BLP and Biba together to make a system → that is, stuffing military security (BLP) and making it commerical
  - Keep development and production separate but give a way to move system from development to production
- **Lipner's integrity matrix model:**
  - Confidentiality levels & categories
  - Integrity levels & categories
  - Just apply BLP and Biba on system
- **Downgrade:** move software from development to production
  - Changing label on object from development to production
  - You won't find this in BLP or Biba
- Shows a way to implement commercial security / integrity but there are better ways

→ QUESTIONS:

1. Are the SD and ID categories in Lipner's model related to each other?

→ No, they are separate from each other. SD is for confidentiality and ID is for integrity.

2. Why is it necessary for system controllers to have to ability to downgrade?

→We need to downgrade in order to give the ability of using production software to system programmers. The BLP rules and the Biba Strict rules don't allow for that so we need to create this hybrid to allow for downgrades.

3. Can system controllers modify development code/test data?

→No, they can read but not modify.

4. What form of tranquility underlies the downgrade ability?

→Weak tranquility underlies the downgrade ability because it allows for change in labels as long as it doesn't go against the "spirit" of either BLP or Biba Strict.

[#6] LECTURE 24

- commercial security has it's own unique concerns so there needs to be another policy
- **Clark-Wilson:**
    - Meta: consistency among various components of the sytem state
    - **Four basic concerns:**
        - Authentication: knowing who each person is who does stuff in the system
        - Audit: modifications logged to record every program executed and by whom
        - Well-formed transactions: you can ony manipulate constrained forms
        - Separation of duty: you need different people carrying out one goal
    - Constrained

JUST KNOW THE BASIC IDEA. DON'T MEMORIZE SLIDE 5 – POLICY RULES.

→ QUESTIONS:

1. What is the purpose of the four fundamental concerns of Clark and Wilson?
→Consistency among the components of the system state.

2. What are some possible examples of CDIs in a commercial setting?
→In terms of banking, bank balances and checks.

3. What are some possible examples of UDIs in a commercial setting?
→In terms of banking, candy from a candy bowl at the bank and the pen you use to sign a check.

4. What is the difference between certification and enforcement rules?
→Certification: given a static state, does it meet all the invariants; making sure that you're in a good state. Enforcement: about operations on a state; is a good state maintained. Certification is about meeting requirements whereas Enforcement is keeping those requirements.

5. Give an example of a permission in a commercial setting.
→In terms of banking, (banker Amy, validate,{checks, money orders}). That is, banker Amy can validate checks and money orders.

[#6] LECTURE 25

- **Very specific commercial concern:** the potential for conflicts of interest and inadvertant disclosure of information by a con sultant or contractor
    - **Ex:** someone goes from working for LG to Samsung. He could give Samsung inside LG information. How do you control the information that gets out?
- **Chinese-wall policy:**
    - Access control confidentiality policy; not really an integrity policy
        - Objects: data about only one company
        - Company groups: all objects about a company
        - Conflict classes: set of company groups that are directly competing
    - **Meta:** subject can access info from any company as long as they never accessed infor from any different company in the same conflict class
    - Permissions change dynamically – access depends on history of past accesses
- Fact

→ QUESTIONS:

1. Why would a consultant hired by American Airlines potentially have a breach of confidentiality if also hired by United Airlines?

→The consultant could give information to United Airlines about American Airlines, both intentionally and inadvertently.

2. In the example conflict classes, if you accessed a file from GM, then subsequently accessed a file from Microsoft, will you then be able to access another file from GM?

→Yes, you can access multiple files in a single conflict class, given that it's from the same company.

3. Following the previous question, what companies' files are available for access according to the simple security rule?

→For the same subject in the previous question, the subject would be able to access GM, Microsoft, and one of the three: Bank of Americ a, Wells Fargo, and Citicorp.

4. What differences separate the Chinese Wall policy from the BLP model?

→The Chinese Wall policy is specifically designed to address a specific concern: conflicsts of interest by a consultant or contractor. The BLP model is very general, though they are both access control policies.

[#6] LECTURE 26

- **Role-based access control (RBAC)**
  - Especially appropriate for Commercial settings
  - Doesn't set individual permissions to every single subject; sets permissions to functions/jobs/roles
    - i.e. president, manager, trainer, teller, auditor, janitor
  - users have a set of **authorized roles** and a set of **active roles** (can be multiple)
- Rules:
  - role assignment: if you don't have an active role, you can't do anything
  - role authorization: your active roles have to be an authorized role
  - transaction authorization: our transactions must be authorized by one of your active roles
- subsume:
  - anyone having role x can do anything role j can do.
  - Ex: a trainer can do all the duties that a trainee can do.
- separation of duty: (checks and balanaces)
  - ex: if you're a teller, you can't be a banker.
- sdf

→ QUESTIONS:

1. What benefits are there in associating permissions with roles, rather than subjects?

→ It's much easier to administer; it is more real-world friendly in that you don't have to litearlly assign specific permissions to every single worker at a bank (for example).

2. What is the difference between authorized roles and active roles?

→Authorized roles are the set of roles you are allowed to fill whereas active roles are the ones you are currently filling. Authorized roles are a superset of active roles.

3. What is the difference between role authorization and transaction authorization?

→Role authorization is making sure that a subject's active roles are in its set of authorized roles whereas transaction authorization makes it so that the subject's transactions have to be authorized by one of its active roles.

4. What disadvantages do standard access control policies have when compared to RBAC?

→It's much easier to administer RBAC and much more flexible than the standard access control policies. Also, permissions are more appropriate to each different organization with the RBAC. Standard access control policies are more confining.

[#6] LECTURE 27

- ACM for any access control system
- Fact
- Fact

→ QUESTIONS:

1. Why would one not want to build an explicit ACM for an access control system?

→Because in realistic systems, most subjects don't have access to most objects. It would be a waste of time to build an explicit, giant ACM.

2. Name, in order, the ACM alternatives for storing permissions with objects, storing permissions with subjects and computing permissions on the fly.

→Storing permissions of objects: access control list. Stroing permissions with subjects: capability-based system. Computing permissions on the fly: you just compute them on the fly using the various rules we have learned thus far (i.e. simple security and *-property). That is, you maintain a set of rules to compute access permissions “on the fly” based on attributes of subjects and objects. 2 Store the permissions with objects. This is called

[#7] LECTURE 28

- Information: any content to be conveyed that is sent in the form of one or more messages
- structure: sender → channel/ transmission medium → reciever
- Fact
- Fact

→ QUESTIONS:

1. What must be true for the receiver to interpret the answer to a “yes” or “no" question?
→The sender has an answer. The receiver belives that the sender has an answer. There must be a channel between the sender and the receiver. The send and receiver have to have some shared knowledge, including an agreed encoding scheme.

2. Why would one want to quantify the information content of a message?
→It affects communication, hardware design, protocol designs, etc. You want to know how much information can be pushed through a channel to get it to work efficiently.

3. Why must the sender and receiver have some shared knowledge and an agreed encoding scheme?
→If the sender and receiver don't have an agreed encoding scheme, the sender can send all the data it wants to the receiver but the receiver won't know what to do with it; the receiver will still be in the dark.

4. Why wouldn’t the sender want to transmit more data than the receiver needs to resolve uncertainty?
→The passage of information should be as efficient as possible. Sending more than necessary will clog up the system and confuse the receiver.

5. If the receiver knows the answer to a question will be “yes,” how many bits of data quantify the information content? Explain.
→There are only 2 answers possible so you only need 1 bit, 1 and 0 representing yes and no. Of course, the receiver and the sender have to be aware of the encoding scheme.

[#7] LECTURE 29

- Fact
- Fact
- Fact

→ QUESTIONS:

1. How much information is contained in each of the first three messages from slide 2?
→n-bits, 4 bits, 7 bits, and 22*8 bits since there are 22 chars and each char is 8 bits long for the last one, assuming that the sender and reciever don't have an agreed upon encoding scheme.

2. Why does the amound of information contained in "The attack is at dawn" depend on the receiver's level of uncertainty?
→Should the reciever and sender have an agreed upon encoding scheme, the message could be contained in less bits (i.e. it would be 4 bits if there was only 16 possible times; it would be 1 bit if the attack could only be at dawn or dusk). Assuming that they don't have an ecoding scheme, it would be 22*8 bits of information, which is far from efficient. It all depends on the reciever's level of uncertainty.

3. How many bits of information must be transmitted for a sender to send one of exactly 16 messages? Why?
→4. There are exactly 16 combinations of 1 ands 0s with 4 bits of code.

4. How much information content is contained in a message from a space of 256 messages?
→8 bits. Log base 2(256) = 8.

5. Explain why very few circumstances are ideal, in terms of sending information content.
→Only in an ideal situation could you reduce certainty in half with each bit transmitted. However, that rarely happens in the real world. You can very well get cases where the reciever doesn't know how many message possibilities there are.

[#7] LECTURE 30

- Fact
- Fact
- Fact

→ QUESTIONS:

1. Explain the difference between the two connotations of the term "bit."
→In a discrete setting, it is a binary digit, meaning a 0 or a 1. In a continuous setting, we're measureing the quantity of information.

2. Construct the naive encoding for 8 possible messages.
→**M1 =** 000; M2 = 001; M3 = 010; M4 = 011; M5 = 100; M6 = 101; M7 = 110; M8 = 111.

3. Explain why the encoding on slide 5 takes 995 + (5 * 5) bits.
→995 times out of 1000, the message will be 0, meaning that it can be represented by 1 bit, which is where the 995 comes from since 995 * 1 = 995. 5 times out of 1000, the message will be a 5-bit encoding, meaning that it would be 5*5. Putting these together, the encoding takes 995+*5*5) = 1020 bits or 1.02 bits per message.

4. How can knowing the prior probabilities of messages lead to amore efficient encoding?
→You can come with an encoding that makes it more efficient to transmit the more probable messages. For example, let's say that we have 2 messages that come up 99.5% of the time out of 20 total messages. Then, we could make a separate encoding for the 2 messages (i.e. 1 bit, 1 for the first message and 0 for the second message). This would make ecoding much more efficient.

5. Construct an encoding for 4 possible messages that is worse than the naïve encoding.
→M1 = 11111111; M2 = 00001111; M3 = 11110000; M4 = 11111111;

6. What are some implications if it is possible to find an optimal encoding?
→It has to be the most efficient, the best possible encoding that we can possibly have.

[#7] LECTURE 31

- Language:
  - Symbol set
  - Refers to results of a series of experiements
  - Ex for coin toss: TTHTHTH
- Encoding properties:
  - Lossless: possible to recover entire original sequence of symbols
  - Uniquely decodable: there must be only one possible decoding; there's not two different meanings
  - Streaming: no breaks in encoding
-

→ QUESTIONS:

1. Name a string in the language consisting of positive, even numbers.
→Drawing from a sack with balls labeled with positive, even numbers ranging from 2 to 8. In this case, a string might be: 2684468222486624.

2. Construct a non-prefix-free encoding for the possible rolls of a 6-sided die.
→ Using naïve encoding: 1: 000, 2: 001, 3: 010, 4: 011, 5: 100, 6: 101.

3. Why is it necessary for an encoding to be uniquely decodable?
→Using naïve encoding: 1: 000, 2: 001, 3: 010, 4: 011, 5: 100, 6: 101.

4. Why is a lossless encoding scheme desirable?
→We want to be able to send the complete encoding so that the reciever gets the whole information; there's no loss information.

5. Why doesn't Morse code satisfy our criteria for encodings?
→If you have three dots in a row, how do you kno if it's one S or three E's? We can't so it doesn't satisfy our criteria for encodings.

[#7] LECTURE 32

- Fact
- Fact
- Fact

→ QUESTIONS:

1. Calculate the entropy of an 8-sided, fair die (all outcomes are equally likely).
→ Entropy = - (8 * (1/8 log (1/8)), given that each side has a 1/8 probability of being landed on.

2. If an unbalanced coin is 4 times more likely to yield a tail than a head, what is the entropy of the language?
→ Entropy = -(4/5 log(4/5) + 1/5 log(1/5))

3. Why is knowing the entropy of a language important?
→ It lets you know theoretically how many bits on average you need to transmit the information/results.

[#7] LECTURE 33

- Fact
- Fact
- Fact

→ QUESTIONS:

1. Explain the reasoning behind the expectations presented in slide 3.

→The probabilities of each toss are indepenedent of each other so you just multiply the individual probabilities. So, HH: (3/4)*(3/4) = 9/16, HT = TH = (3/4)*(1/4) = 3/16, and TT = (1/4)*(1/4) = 1/16. This means that you'll likely get 9 HH, 3 HT, 3 TH, and 1 TT in a case where we flip the coin 32 times.

2. Explain why the total expected number of bits is 27 in the example presented in slide 4.

→Total bits = (count_of_result1 * bits_in_code1) + (count_of_result2 * bits_in_code2) + (count_of_result3 * bits_in_code3) + (count_of_result4 * bits_in_code4) = (9*1) + (3*2) + (3*3) + (1*3) = 9+6+9+3 = 27.

3. What is the naive encoding for the language in slide 5?

→1: 000, 2: 001, 3: 010, 4: 011, 5: 100, 6: 101.

4. What is the entropy of this language?

→

1,2: 6/20
3,4: 3/20
5,6: 1/20
Thus, entropy: -((2*(6/20 log 6/20)+ (2*(3/20 log 3/20) + (2*(1/20 log 1/20))

5. Find an encoding more efficient than the naive encoding for this language.

→ 1: 0, 2: 10, 3: 110, 4: 1110, 5: 11110, 6: 11111

6. Why is your encoding more efficient than the naive encoding?

→1 and 2 are rolled most often so they're represented by 1 bit and 2 bit encodings. 3 and 4 are rolled next most often so they are represented by 3 and 4 bits respectfully. The most probable rolls are thus represented by less bits, meaning that it will be more efficient in the long run.

[#8] LECTURE 34

- **Entropy:** provides a bound on coding efficiency
- **Entropy of language:** measure of the most efficient possible encoding of a language
- **Entropy of message source:** amount of information content that the source can produce in a given period; how much info you can send through a channel
    - **Ex:** a grandma and a teenager telling the same story but in different lengths → grandma takes forever with many tangents, teenager tells it as short as possible
- Any channel can send any arbitrary amount of info given enough time and unbounded buffering capacity
- Shannon's Theorems:
    - show that it is always possible to approach that limit arbitrarily closely

→ QUESTIONS:

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

→ According to the Fundamental Theorem of the Noiseless Channel, It is impossible to transmit at an average rate greater than C/h, assuming that a language has entropy h (bits per symbol) and a channel can transmit C bits per second.

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

→ According to the Fundamental Theorem of a Noisy Channel, covert channels in the system cannot be dismissed with the argument that they are noisy and hence useless because you can always get the message through by finding a more redundant encoding. That is, if there is any bandwidth at all, you can eventually find an encoding to get your message across the channel, though you will have to add some redundancy.

[#8] LECTURE 35

- entropy of natural language (i.e. English)
- entropy of English is less than standard encoding
- there's a lot of redundancy in English encoding
    - you can remove letters out of passages and it'll still make sense but you can't just remove them when you encode English.
- Zero-order model:
    - All characters are equally likely
- Fact

→ QUESTIONS:

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

→Assume that all digits are equally likely so the entropy of the language would be –(log(1/10)).

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

→No one actually knows what the exact entropy of any given natural language is (i.e. English) because there are so many possibilities and factors. Computing the entropy of a natural language requires very complex models so you can only really get an estimate.

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

→In zero-order, All characters are assumed to be equally likely. In first-order, we assume that all symbols are independent of one another but follow the given probabilities. In second-order, we assume that all symbols are dependent on one another (i.e. likelihood of digrams, trigrams, etc.) and calculate the entropy.

[#8] LECTURE 36

- Entropy depends on how much the receiver knows
- Fact
- Fact

→ QUESTIONS:

1. Why are prior probabilities sometimes impossible to compute?

→Knowing exactly what the exact entropy is happens to be for any observer is some complicated likelihood of some events and we can't really compute that because a lot of prior probabilities depend on too many factors.

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

→ Entropy is relative to a particular observer because information content of a message depends on the state of knowledge of the receiver.

3. Explain the relationship between entropy and redundancy.

→Entropy can be used to measure the amount of "redundancy" in the encoding. If the information content of a message is equal to the length of the encoded message, there is no redundancy.

[#9] LECTURE 37

- What we'll learn about Cryptography:
  - Key concepts
  - How it's used in security
  - How effective it is
- Fact
- Fact

→ QUESTIONS:

1. List your observations along with their relevance to cryptography about Captain Kidd's encrypted message.
→You should ask yourself what the underlying language is in the first place. Is it English? Are there any clues? One clue might be that there's a goat head on the paper, which is a pun to Captain Kidd's name. This pun only works in English so we know that the source language is English. →It's probably a simple substitution text algorithm; it's not modern ecryption.

2. Explain why a key may be optional for the processes of encryption or decryption.
→Keyless ciphers do not use keys. An example in which you might not need a key would be in a keyless transposition cipher where positions held by untils in the plaintext are simply shifted so there is no key necessary. Sometimes, the cipher isn't complex enough to require a key.

3. What effect does encrypting a file have on its information content?
→The information content should be preserved. Encrypting should preserve the information content so that the receiver could get the information but to hide it. There should be no change to the file's information content.

4. How can redundancy in the source give clues to the decoding process?
→Redundancy is the enemy of secure encryption because it provides leverage to the attacker in that they can use redundancy to figure out what the key is or what the original message was. We want ecryption to obscure the meaning of text.

[#9] LECTURE 38

- Fact
- Symmetric encryption:
    - Same key for encryption and decryption
- Asymmetric encryption:
    - Different keys
- Keyless cipher:
    - Doesn't use a key
-

→ QUESTIONS:

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

→D(E(D(E(P)))) = D(E(P)) = P

2. Rewrite the following in its simplest form: D(E(E(P,KE),KE),KD).

→ D(E(E(P,KE),KE),KD) = D(E(C,KE),KD).

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

→Recognizing patterns may lead to the cryptanalyst finding out the key. There is something called traffic analysis (i.e. a particular entity sends more entity when a crisis is going on). You might infer the traffic without knowing anything about the content which might give you some clues about the scenario.

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

→For example the frequency of certain symbols in a specifc language would possibily allow you to use zero, first, second and third-order models.

[#9] LECTURE 39

- Encryption algorithm is Breakable:
    - If analyst can recover plaintest given time and data
    - Most are breakable since anaylist can try all keys systematically but it's not very feasile
- Fact
- Fact

→ QUESTIONS:

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

→ Most encryption algorithms are breakable since the analyst can try all keys systematically. However, it isn't very feasible to have an analyst try all the keys; an exhaustive search is generally very naïve.

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

→ Many ciphers use a n -bit string as key. Given a small number of plaintext/ciphertext pairs encrypted under key K, you can use brute force efficiently (expected on time on the order of 2^(n-1)). You're just doing a linear search on that so on average, you find the correct one halfway there so that's where 2^(n-1) comes from.

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

→ Substitution and transportion are the simplest building blocks of encryption. It might seem that these are too naive to be effective. But almost all modern commercial symmetric ciphers use some combinati on of substitution and transposition for encryption.

4. Explain the difference between confusion and diffusion.

→Confusion is transforming information in plaintext so that an interceptor cannot readily extract it whereas difficusion is spreading the information from a region of plaintext widely over the ciphertext.

5. Is confusion or diffusion better for encryption?

→ Substitution tends to be good at confusion; transposition t ends to be good at diffusion. Almost all modern commercial symmetric ciphers use some combinati on of substitution and transposition for encryption so rather than think of either of them as better than the other, they should be used together.

[#10] LECTURE 40

- Fact
- Fact
- Fact

→ QUESTIONS:

1. What is the difference between monoalphabetic and polyalphabetic substitution?
→They are both tyes of substitution ciphers. However, if this is done uniformly this is called a monoalphabetic cipher or simple substitution cipher and if different substitutions are made depending on where in the plaintext the symbol occurs, this is called a polyalphabetic substitution.

2. What is the key in a simple substitution cipher?
→It is however you specify the mapping so it might be a table that tells you the mappings.

3. Why are there k! mappings from plaintext to ciphertext alphabets in simple substitution?
→It's an injection (1-1 mapping) of the alphabet itself into itself or another alphabet so it has to have k! mappings.

4. What is the key in the Caesar Cipher example?
→It's how many positions you shift.

5. What is the size of the keyspace in the Caesar Cipher example?
→English has 26 letters so it's 26, or 25 depending on how you look at it.

6. Is the Caesar Cipher algorithm strong?
→No. It's not a particuarlly weak algorithm but you don't have to try all the keys before you get the right one.

7. What is the corresponding decryption algorithm to the Vigenere ciphertext example?
→The Vigenere example uses a symmetric-key algorithm using a block, stream cipher. It is also an example of a polyalphabetic substitution.

[#10] LECTURE 41

- Fact
- Fact
- Fact

→ QUESTIONS:

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

→You have 3 symbols/letters and there are 26 letters in the English alphabet so there are 26*26*26 decryptions possible. Thus, 26^3 = 17576.

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

→We know that there are two Y's so using simple substitution cipher, we know that there are 26*25 possibilities, thus reducing the search by a factor of 27.

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

→ A perfect cipher would be one in which no reduction of the search space is possible, even given access to the ciphertex t and algorithm. Given the defitnion, yes, a perfect cipher is possible. We know that a perfect cipher is possible by the one-pad.

[#10] LECTURE 42

- Fact
- Fact
- Fact

→ QUESTIONS:

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

→ Every possible plaintext could be the pre-image of that ciphertext under a plausible key. Therefore, no reduction of the search space is possible.

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

→You have to know absolutely nothing about the key other than that it is random, otherwise it wouldn't be a perfect ecryption. If for example you knew that the key had even parity, then you could work backwards and eliminate half of the plaintext. Again, you must know nothing about the key other than that it was generated randomly.

3. Explain the key distribution problem.

→The sender and the receiver both need a key but how do you get it to each other securely? If sender and receiver already have a secure channel, why do they need the key and If they don't, how do they distribute the key securely? It's a problem.

[#10] LECTURE 43

- Fact
- Fact
- Fact

→ QUESTIONS:

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

→ A disadvantage is that such ciphers are considerably more laborious and error prone than simpler ciphers. Also, they are regarded as a building block for encryption. Transposition ecryptions aren't very strong so most commercial algortihsm use combinations, though a bombination is not necessarily stronger than either cipher individually; it may even be weaker.

[#10] LECTURE 44

- Fact
- Fact
- Fact

→ QUESTIONS:

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

2. Describe the difference between key distribution and key management.
→Key distribution is how do we convey keys to those who need them to establish secure communication whereas key management is given a large number of keys, how do we preserve their safety and make them available as needed.

3. If someone gets a hold of Ks, can he or she decrypt S's encrypted messages? Why or why not?
→Each subject has a pubically disclosed key Ks. Anyone can use it to encrypt but it can't be used to decrypt because it is asymmetric encryption, meaning that the user would need another key to use separately for decryption.

4. Are symmetric encryption systems or public key systems better?
→Symmetric key algorithms are much faster computationally than asymmetric algorithms as the encryption process is less complicated. Symmetric key systems are less expensive as well so if one had to be chosen, a symmetric ecryption system would technically be "better" on a general level. Symmetric encryption remains the work horse of commercial cryptography, with asymmetric encryption playing some imp ortant special functions

[#10] LECTURE 45

- Fact
- Fact
- Fact

→ QUESTIONS:

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

→Block ciphers have high difficusion (information from one plaintext symbol is diffused into several ciphertext symbols) and immunity to tampering (difficult to insert symbols without detection).

2. What is the significance of malleability?

→It's a bad thing for an encryption algorithm and you don't want it. It means that it allows transformations on the ciphertext produce meaningful changes in the plaintext. Given that your attacker makes a change, you're not even going to notice that there was any change but when you decode it, the message might have changed completely. If you change even one bit in a file, it is liable to change the whole result of the decryption of the block.

3. What is the significance of homomorphic encryption?

→ Homomorphic encryption is a form of encryption where a specific algebraic operation performed on the plaintext is equivalent to another (possibly different) algebraic operation performed on the ciphertext. The signifigance is that homomorphic encryption schemes are malleable by design. For example, the homomorphic property of various cryptosystems can be used to create secure voting systems, collision-resistant hash functions, and private information retrieval schemes.

[#11] LECTURE 46

- Fact
- Fact
- Fact

→ QUESTIONS:

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

→ Confusion means that the key does not relate in a simple way to the ciphertext. In particular, each character of the ciphertext should depend on several parts of the key. addRoundKey uses confusion because it doesn't just relate in a simle way to the key. It does this: XOR the state with a 128-bit round key derived from the original key K by a recursive process.

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

→ Diffusion means that if we change a character of the plaintext, then several characters of the ciphertext should change, and similarly, if we change a character of the ciphertext, then several characters of the plaintext should change. The mixColumns step uses diffusion. What it does is for each column of the state, it replaces the column by its value multiplied by a fixed square matrix of integers.

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

→ Inverting the MixColumns step requires multiplying each co lumn by a fixed array so for that reason, decryption typically takes longer than encryption.

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

→AES uses blocks to take input in fixed size blocks. A 128-bit block is arranged as a 4 × 4 array of bytes called the “state,” which is modified in place in each round. The key is also arranged as a 4 × n array of bytes, and is initially expanded in a recursive process into r + 1 128-bit keys, where r is the number of rounds. AES uses 10, 12, or 14 rounds for keys of 128, 192, and 256 bits, respectively. Each round consists of four steps: subBytes, shiftRows, mixColumns, and addRoundKey.

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

→ More rounds means more security against cryptanalysis, simply, since there is more confusion and diffusion

[#11] LECTURE 47

- Fact
- Fact
- Fact

→ QUESTIONS:

1. What is a disadvantage in using ECB mode?

→ A naive use of encryption as in Electronic Code Book leaves too much regularity in the ciphertext. That is, the problem with ECB is that identical blocks in the plaintext will yield identical blocks in the ciphertext. This is a problem for plaintext with frequent repeats, such as internet packet traffic.

2. How can this flaw be fixed?

→ To solve the problem of EBC, do something to "randomize" bloc ks before they're encrypted. That is, you need to use CBC (Cipher Block Chaining).

3. What are potential weaknesses of CBC?

→ Observed changes (an attacker able to observe changes to ciphertext over time will be able to spot the first block that changed) and content leak (If an attacker can find two identical ciphertext blocks, he can derive information about two plaintext blocks).

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

→ In block encryption modes (like ECB and CBC), the point is to generate ciphertext that stores the message in encrypted but recoverable form. However, in key stream generation modes the cipher is used more as a pseudorandom number generator. The result is a key stream that can be used for encryption by XORing with a message stream. Decryption uses the same key stream.

[#11] LECTURE 48

- Fact
- Fact
- Fact

→ QUESTIONS:

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

→The decryption key (aka the private key). The idea behind publick key systems is to use a publicly disclosed key to encrypt and a secret key to decrypt. This drastically reduces the number of keys that have to be protected.

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

→ The basis of any public key system is the identification of a function that is easily computed, but difficult to invert without additional information. This is called a one-way function. This is critical to public key systems because the system allows anyone with the public key (so basically everyone) can encrypt but only you can decrypt with the private key.

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

→Public key systems largely solve the key distribution problem because they remove the need for the receiver and the sender to agree on a joint key, thereby removing the need to share a key securely with each other.

4. Simplify the following according to RSA rules: {{{P}K−1}K}K−1.

→{{{P}K−1}K}K−1 = {P}K−1. So it's the decryption of P.

5. Compare the efficiency of asymmetric algorithms and symmetric algorithms.

→ Asymmetric algorithms are generally much less efficient than symmetric algorithms. Asymmetric algorithms generally cost more as well.

[#11] LECTURE 49

- Fact
- Fact
- Fact

→ QUESTIONS:

1. If one generated new RSA keys and switched the public and private keys, would the algorithm still work? Why or why not?
→Yes, you can use either for encryption and decryption just because it's the way it is symmetrically designed. RSA is symmetric in the use of keys.

2. Explain the role of prime numbers in RSA.
→ In RSA, this asymmetry is based on the practical difficulty of factoring the product of two large prime numbers, the factoring problem.

3. Is RSA breakable?
→ The algorithm is theoretically very secure, but has practic al weaknesses. Thus, it is definitely breakable.

4. Why can no one intercepting {M}Ka read the message?
→Only A has the key which will allow the decryption so we get privacy. Encryption with the public key is a privacy transformation, but not an authenticity transformation.

5. Why can't A be sure {M}Ka came from B?
→You can't get authentication because anyone could have sent the message; it didn't have to be from B. Encryption with the public key is a privacy transformation, but not an authenticity transformation.

6. Why is A sure {M}K−1 b originated with B?
→Because no one but b has that private key that b has so we get authentication. Encryption with the private key is an authenticity transformation, not a privacy transformation.

7. How can someone intercepting {M}K−1b read the message?
→It might be the case that anybody has B's key. Encryption with the private key is an authenticity transformation, not a privacy transformation.

8. How can B ensure authentication as well as confidentiality when sending a message to A?
→ A public key encryption can be used for authenticity or for privacy but not both at once. However, in other public key systems, you typically need two pairs of k eys: one pair for privacy and the other pair for "signing" (authen ticity).

[#11] LECTURE 50

- Fact
- Fact
- Fact

→ QUESTIONS:

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

2. What is the key difference between strong and weak collision resistance of a hash function.
→A function f is weak collision resistant if, given an input m1 , it is hard to find m2 != m1 such that f(m1) = f(m2). Differently, a function f is (strong) collision resistant if it is hard to find two messages m1 and m2 such that f(m1) = f(m2).

3. What is the difference between preimage resistance and second preimage resistance?
→A function f is preimage resistant if, given h , it is hard to find any m such that h = f(m) whereas a function f is second preimage resistant if, given an input m1 , it is hard to find m2 != m1 such that f(m1) = f(m2).

4. What are the implications of the birthday attack on a 128 bit hash value?
→It's almost certain that there's always going to be at least two individuals with birthdays on the same day. This means that you'll have to look at 1.25*(2^64) values before you get a collisions.

5. What are the implications of the birthday attack on a 160 bit hash value?
→ This means that you'll have to look at 1.25*(2^80) values before you get a collisions.

6. Why aren't cryptographic hash functions used for confidentiality?
→ It's usually used for integrity. we usually use encryption to conceal the contents of an object, i.e., to protect confidentiality. However, in some cases integrity is the desired result. In a secure communications system, the correct transmission of messages may override confidentiality concerns.

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

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

[#11] LECTURE 51

- Fact
- Fact
- Fact

→ QUESTIONS:

1. For key exchange, if S wants to send key K to R, can S send the following message: {{K}KS−1}K−1R ? Why or why not?
→Answer

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?
→Answer

3. Is {{{K}KS−1}KR}KS equivalent to {{K}K−1 S}KR?
→Answer

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

[#11] LECTURE 52

- Fact
- Fact
- Fact

→ QUESTIONS:

1. What would happen if g, p and gamodp were known by an eavesdropper listening in on a Diffie-Hellman exchange?
→Answer

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

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

[#12] LECTURE 53

- **Ex: a check (properties it should have???)**
  - check authorizes transaction
  - signature confirms authenticitiy
  - if forged signature, third party can come in to check it out
  - check can't be altere; easy to see if it has been altered
  - signature can't easily be reused because it's literally a part of the check itself
- **digitial signature properties:**
  - acts a lot like physical signatures
  - *S sends signature f(S,M) to R*
  - **unforgeable:** difficult for anyone but S to produce the signature
  - **authentic:** R can verifty that S signed M
  - **no repudiation:** S can't deny signing it
  - **tamperproof:** M can't be modified after being sent
  - **not reusable:** signature can't be detached and reused for another message; signature bound to document
- suited for public key systems (particularly RSA)
  - **unforgeable:** only S has its private key and M's inner encryption is done by the private key
  - **authentic:** R can verifty that S signed M because R's public key is available to everyone
  - **no repudiation:** S can't deny signing it unless S gave away its private key to someone else to use
  - **tamperproof:** the outer layer of encryption is R's public key so no one but R can strip off that layer
  - **not reusable:** signing is the encryption of M itself so you can't remove
-

→ QUESTIONS:

***** LECTURE 53 *****

1. Why is it important for a digital signature to be non reusable?

→Should a digital signure be reusable, it would ruin three of the properties that it needs: unforgeable, authentic, and no repudiation. That is, if a digital signure was reusable, it would be easy for anyone to produce the signature, R wouldn't be able to verify that S signed the document, and S could deny producing the signature.

2. Why is it the hash of the message typically signed, rather than the message itself?

→It's because public key encryption is expensive to apply and the message might be arbitrailiy long but the hash will be a fixed, finite, short value.

3. What assurance does R gain from the interchange on slide 4?

→It is unforgeable, authentic, no repudiation, tamperproof, and not reusable. It holds all the properties you would want from a digital signature.

[#12] LECTURE 54

- problem:
    - when you're on a distributed system and you're dealing with strangers, how do you know that the key they're sending you is really theirs?
    - **Certificate:** a way for one party vouching for a bounding between an identity (person) and a public key
- Need for trust:
    -
- Fact

→ QUESTIONS:

***** LECTURE 54 *****

1. What is the importance of certificate authorities?
→We need some way of knowing that there's a binding between a public key and a user's identity. It's some party saying that you can trust the information given.

2. In the example on slide 5, why does X sign the hash of the first message with its private key?
→X is vouching for Y so X can't use its public key because that, in turn, would need to be vouched for by another certificiate. X is the certifying authority and as such, X must use its private key.

3. Why is it necessary to have a hash of Y and Ky?
→We need to make sure that data items Y and K Y were not altered or corrupted. Plus, it's much less costly to use hashes.

4. What would happen if Z had a public key for X, but it was not trustworthy?
→The entire system would be flawed because then you would need a certificiate for the vouching authority (i.e. X), to validiate X's trust. Then, there would be no point in having X vouche for Y.

[#12] LECTURE 55

- Fact
- Fact
- Fact

→ QUESTIONS:

***** LECTURE 55 *****

1. What happens at the root of a chain of trust?
→Ideally, the chain is rooted at some unimpeachable authority. To be useful the chain must be rooted in a trusted author

2. Why does an X.509 certificate include a "validity interval"?
→The validity interval serves to measure the start and end times for validity. It tells you how long a certificate is valid for. If a certificiate is expired, there's no point in trusting it.

3. What would it mean if the hash and the received value did not match?
→If the two are the same, it's pretty certain that the certificiate is correct. If the hash and the received value did not match, it would invalidiate the certificate and furthermore that the data could have been altered.

[#13] LECTURE 56

- Fact
- Fact
- Fact

→ QUESTIONS:

***** LECTURE 56 *****

1. What are some protocols previously discussed?
→A protocol is a structured dialogue. Some protocols we have previously discussed are public key encryption and covert channels.

2. What may happen if one step of a protocol is ignored?
→There may be miscommunication and the message might be misinterpreted.

3. Why must the ciphers commute in order to accomplish the task in slide 4?
→We need to reach inside the outer encryption and take the inner encryption off but most algorithms don't allow this. We need it to commute (that is, do the encryption in any order) so that we can complete the steps.

4. Describe how an attacker can extract M from the protocol in slide 6.
→ In step 3, the two applications of Ka "cancel out," leaving (M xor Kb), which B can easily decrypt with his own key Kb. Also, an attacker who stores the three messages can XOR combinations of them to extract any of M, Ka , and Kb. To extract M, M XOR 0 is just M.

5. Describle how an attacker can extract Ka from the protocol in slide 6.
→ You assume that the attacker has all of the messages that were sent. By XOR'ing them together, you can get any of M, Ka, or Kb. That's because if you XOR something with itself, gives 0. Ka XOR 0 is just Ka.

6. Describle how an attacker can extract Kb from the protocol in slide 6.
→ You assume that the attacker has all of the messages that were sent. By XOR'ing them together, you can get any of M, Ka, or Kb. That's because if you XOR something with itself, gives 0. Kb XOR 0 is just Kb.

7. Why are cryptographic protocols difficult to design and easy to get wrong?
→It's difficult designing them to work so that they work robustly and efficiently yet have appropritate privacy due to the distributed nature of the environment. It's also difficult to define what constitutes an attack.

[#13] LECTURE 57

- Fact
- Fact
- Fact

Test:

- a protocol example slide: given equations, tell what is going on
-

needham Schroder slide

"TWO QUESTIONS TO ASK OF ANY STEP IN A PROTOCOL"

- don't memorize prototocals (needham0schroder)
- if he'll test on protocol, he'll give it to you
- he wants to know if you can interpret what's happening and if you can explain it

"BELIEF LOGICS: BAN"

- go through examples but don't memorize the details.
- He doesn't want logical formulas so just write it out in english

→ QUESTIONS:

***** LECTURE 57 *****

1. Explain the importance of protocols in the context of the internet.

→Almost everything that occurs on the Internet occurs via a protocol. The internet itself needs communication between two or more parties to carry out functions. This applies to when you move a file, when you e-mail someone, etc.

1. Explain the importance of cryptographic protocols in the context of the internet.

→A cryptographic protocol is a protocol using cryptographic mechanisms to accomplish some security related function. You want to have secure exchanges on the internet. You want to protect your assets so you need cryptographic prototcols.

3. What are the assumptions of the protocol in slide 6?

→Thee assumptions are that there is a public key infastructre in place and the public keys are trustworthy. However, note that there is typically no guarantee that B receives the message, or is even expecting the message.

4. What are the goals of the protocol in slide 6?

→Unicity (secret shared by exactly two parties), integrity (message arrived unmodified), and non-repudiation both ways (sender/receiver can't deny sending/receiving).

5. Are the goals of the protocol in slide 6 satisfied? Explain.

→Answer

6. How is the protocol in slide 6 flawed?

→

[#13] LECTURE 58

- Fact
- Fact
- Fact

→ QUESTIONS:

***** LECTURE 58 *****

1. Why is it important to know if a protocol includes unnecessary steps or messages?

→ Like with the other items in the set of protocol questions, you want to know your protocol's faults. As has been said before, protocols are extremely difficult to design and easy to get wrong. You want to know if a protocol includes unnecessary steps or messages because that could be a potential weakness for the protocol; that portion could be susceptible to an attack.

2. Why is it important to know if a protocol encrypts items that could be sent in the clear?

→ As like the first question, this one is important as well because it is an issue in protocol implementation that you should be aware of when designing one. The answer is the same as above. Like with the other items in the set of protocol questions, you want to know your protocol's faults. As has been said before, protocols are extremely difficult to design and easy to get wrong. You want to know if a protocol includes unnecessary steps or messages because that could be a potential weakness for the protocol; that portion could be susceptible to an attack.

[#14] LECTURE 59

- Fact
- Fact
- Fact

→ QUESTIONS:

***** LECTURE 59 *****

1. Why might it be difficiult to answer what constitutes an attack on a crypto-graphic prototocol?

→It's difficult to give it a set definition. It has too many dependencies and it really does change on a case-by-case basis. Just one of the many questions we need to ask ourselves is if it is possible to impersonate one or more of the parties.

2. Describe potential dangers of a replay attack.

→A replay attack is when the attacker records messages and replays them at a later time. A potential danger is that the attacker confuses the parties. Another potential danger is that the attacker could record confidential information from parties and use them in the future.

3. Are there attacks where an attacker gains no secret information? Explain.

→Yes, for example, there's the interleaving attack in which the attacker injects spurious messages into a protocol run to disrupt or subvert it. The attacker's goal isn't to gain secrets but to disrupt/subvert the protocol.

4. What restrictions are imposed on the attacker?

→The designer of a protocol should assume that an attacker can access all of the traffic and interject his own messages into the flow. However, there are limitations on what the attackers can accomplish and what reprecusions those attacks can cause. One restriction, for example, is assuming that the attacker can't write a message that's encrypted with a key he doesn't have.

5. Why is it important that protocols are asynchronous?

→The protocol should be robust in the face of such a determined and resourceful attacker. Due to the distributed nature of the system, protocols must be highly asynchronous. And furthermore, the distributed nature of the system means that no-one but the initiator knows the protocol is running until they recei ve their first message.

[#14] LECTURE 60

- Fact
- Fact
- Fact

→ QUESTIONS:

***** LECTURE 60 *****

1. Would the Needham-Schroeder protocol work without nonces?

→The protocol would still work, technically, but it would be flawed without the gaurentee that the messages are fresh and not a replay from an earlier exchange. It would also be unfair to call it a Needham-Schroeder protocol still if the nonces were taken out because the nonces are a vital factor in the protocol.

2. For each step of the NS protocol, answer the two questions on slide 5.

→ Step1:
(Sender's message) Hey, S! I'm A and I want to talk to B, so generate a new key for us. And by the way, here's a nonce that you can use in subsequent messages so we'll be sure that you're responding to this request.
(Receiver believes) A wants to talk to B, so I need to generate a new session key and get it to them. I should use Na in the response so that they'll know it's fresh.

Step2:
(Sender's message) S generates an appropriate session key Kab for use by A and B and sends it to A in a message encrypted with their shared key Kas.
(Receiver believes) S has set up our shared key and it's valid and timely because of the nonce. Nice.

Step3:
(Sender's message) A relays the new session key to B.
(Receiver believes) A send me the new session key and it's valid and timely because of the nonce.

Step4:
(Sender's message) B sends an acknowledgement to A.
(Receiver believes) B got my message so yay! And it's valid because B sent me a new nonce. Cool.

Step5:
(Sender's message)
(Receiver believes)

[#14] LECTURE 61

- Fact
- Fact
- Fact

→ QUESTIONS:

***** LECTURE 61 *****

1. As in slide 5, if A's key were later changed, after having Kas compromised, how could A still be impersonated?

→Answer

2. Is it fair to ask the question of a key being broken?

→Answer

3. How might you address these flaws if you were the protocol designer?

→Answer

[#14] LECTURE 62

- Fact
- Fact
- Fact

→ QUESTIONS:

***** LECTURE 62 *****

1. What guarantees does Otway-Rees seem to provide to A and B?

→It is decently sychronized in that both parties know that the exchange of keys will happen before hand pretty much, whereas in Needham Schroeder B receives a key out of nowhere.

2. Are there guarantees that Needham-Schroeder provides that Otway-Rees does not or vice versa?

→Needham Shroeder provides authentication back to A that B has already received the message, whereas in OR, B doesn't know that A has received the key or not. The point of view of idealization is the the protocol has run and now you're asking, in hindsight, what was the purpose of each step.

3. How could you fix the flawed protocol from slide 4?

→The trick is preventing the encryption and decryption steps from being right next to each other. You can change the protocol to prevent that in a number of ways.

[#14] LECTURE 63

- Fact
- Fact
- Fact

→ QUESTIONS:

***** LECTURE 63 *****

1. Why is the verification of protocols important?

→Protocols are crucial to the Internet; it would be great to ge t them right. Protocols can be notoriously difficult to get correct. Flaws ha ve been discovered in protocols published many years before. It would be nice to be able to reason formally about protocol correctness, which is what verification is. You want to make sure that your protocol is as valid as possible, and formalize it at that.

2. What is a belief logic?

→A belief logic is a formal system for reasoning about beliefs . Any logic consists of a set of logical operators and rules of inference. Belief logics allow reasoning about what principals within the protocol should be able to infer from the messages they see. Allows abstract proofs, but may miss some important flaws.

3. A protocol is a program; where do you think beliefs come in?

→Protocols are crucial to the Internet; it would be great to ge t them right. For this reason, we need to have reasoning behind protocols. In turn, reasoning rigorously about protocols requires some way of formalizing their behavior and properties. Belief logics is such an approach

[#14] LECTURE 64

- Fact
- Fact
- Fact

→ QUESTIONS:

***** LECTURE 64 *****

1. What is a modal logic?
→Answer

2. Explain the intuition behind the message meaning inference rule.
→If A believes (A share(K) B) and A sees { X } K then A believes(B said X).

3. Explain the intuition behind the nonce verification inference rule.
→If A believes X is fresh and A believes B once said X, then A believes B believes X.

4. Explain the intuition behind the jurisdiction inference rule.
→If A believes B has jurisdiction over X and A believes B believes X, then A believes X.

5. What is idealization and why is it needed?
→To get from protocol steps to logical inferences, we have a pr ocess called idealization . This attempts to turn the message sent into its intended semantics. One purpose of idealization and a reason why it is needed is to omit parts of the message th at do not contribute to the beliefs of the recipients.

[#14] LECTURE 65

- Fact
- Fact
- Fact

→ QUESTIONS:

***** LECTURE 65 *****

1. Why do you think plaintext is omitted in a BAN idealization?
→Answer

2. Some idealized steps seem to refer to beliefs that will happen later in the protocol. Why would that be?
→Answer

3. One benefit of a BAN proof is that it exposes assumptions. Explain that.
→Answer