DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.033—Computer System Engineering Handout #23—April 7, 1999

Practice Quiz 2

These quizzes contain problems which have been given for the past five years. The solutions will be handed out separately. As before, one old question will be included on this year's quiz.

 Name ____________________

Problem 1. *(25 points)* Lucifer is determined to figure out Alice's password by a brute-force attack. From watching her log in he knows that her password is eight characters long and all lower-case letters. He sets out to try all possible combinations of eight lower-case letters.

a. Assuming he has to try about half the possibilities before he runs across the right one, one trial can be done in one machine cycle, and he has a 60 MHz computer available, about how long will the project take?

b. How long will it take if Alice chooses an eight-character password that includes upper- and lower-case letters, numbers, and special characters? There are 78 characters available for her to choose from.

c. Suppose processors continue to get faster, improving by a factor of three every two years. How long will it be until Alice's new password can be cracked as easily as her old one?

Problem 2. *(24 points)* Louis Reasoner is fascinated with the discovery that some encryption algorithms are *commutative*.

For this problem we use the usual notation: A message M that is encrypted with key k is written $\{M\}^{k}$. A commutative scheme has the interesting property that for every message and every pair of keys k1 and k2, $\{\{M\}^{k1}\}^{k2} = \{\{M\}^{k2}\}^{k1}$. That is, you get the same result no matter which order two encryptions with different keys are done.

Louis has proposed that Alice, in San Francisco, and Bob, in Boston, use the following scheme for secure delivery of messages between their computers, which are connected via the Internet:

Alice chooses a never-before-used (nonce) encryption key, Ka, encrypts her message M, and sends the result, $\{M\}^{Ka}$, to Bob. Bob chooses another nonce encryption key, Kb, encrypts the already-encrypted message to produce $\{\{M\}^{Ka}\}^{Kb}$ and sends the doubly-encrypted result back to Alice. By commutativity, this message is identical to $\{\{M\}^{Kb}\}^{Ka}$, which is a message that Alice can decrypt with her nonce Ka. She does so, revealing $\{M\}^{Kb}$. This result she sends back to Bob, who can now decrypt it with his nonce Kb to reveal M.

The appealing thing about this scheme is that Alice and Bob did not have to agree on a secret key in advance. Louis calls this the "No-Prior-Agreement" protocol.

a. Is it possible for a passive intruder (that is, one who just listens to the encrypted messages) to discover M? If so, describe how. If not, explain why not.

 Name ______________________

b. Is it possible for an active intruder (that is, one who can also insert, delete, or replay messages) to discover M? If so, describe how. If not, explain why not.

Problem 3. *(10 points)* One way of speeding up a name resolution system is to implement a cache that remembers recently looked-up {name, object} pairs. *Synonyms* are multiple names for the same object. What problems do synonyms pose for cache designers?

Problem 4. *(15 points)* Alyssa is trying to organize her notes on virtual memory systems, and it occurred to her that virtual memory systems can usefully be analyzed as naming systems. She has gone through her 6.033 notes and made a numbered list of some technical terms from the lectures on naming systems; that list is on the right, below. She then listed some mechanisms found in virtual memory systems on the left. But she isn't sure which naming concept goes with which mechanism. Help Alyssa out by placing in each blank on the left the number of the term on the right that best applies to this mechanism. Hint: no number needs to be used twice.

____ a. page map

____ b. virtual address

____ c. physical address

____ d. a TLB entry

____ e. the processor register that points to the page map

1. Search path
2. Naming network
3. Closure
4. Object
5. Name
6. Context
7. None of the above

Problem 5. *(25 points)* The network level of the Internet names hosts with 4-byte *host addresses*. These names are assigned regionally, with all hosts in one major region receiving host addresses for which the first byte is the same. A region is divided into sub-regions in which all host addresses have the same value for the second byte, and so on. This hierarchical naming scheme is used by the routing system to reduce the size of the tables it must maintain.

a. Explain briefly how hierarchical naming of host addresses might reduce the size of routing tables.

b. Ben Bitdiddle is thinking about designing a system for wireless attachment of portable computers to the Internet. He has decided that the basic architecture should consist of a number of fixed base stations, each of which will communicate by radio with those portable computers in its area. He plans to make each base station a sub-sub-region, using the third byte of the Internet address to identify it. Alyssa points out that this plan limits the number of portables that can simultaneously work with one base station. The base station uses up one of the host addresses for its sub-sub-region. How many portables can work with one base station?

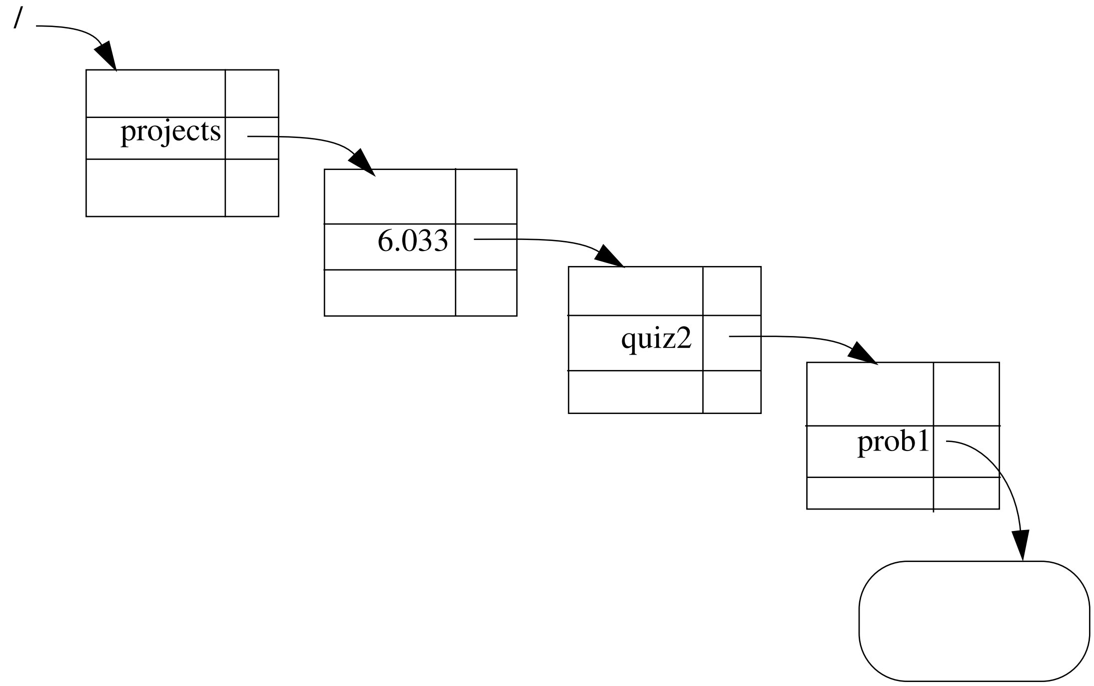
c. Ben proposes not to permanently assign host addresses to portable computers. Instead, when a portable is switched on, it will ask some base station to dynamically assign a host address for it to use. When dynamic address assignment is used, what else has to happen in order for other computers to initiate communication with a portable?

d. Ben wants his design to work with the portable computer mounted on the handlebars of his bicycle. The base stations he will use were designed for cellular phones, so they have the ability at the physical transport level to transfer (or "handoff") a conversation from one base station to another as the signal strength changes. But in Ben's design a handoff means that the host address of the portable must change. Ben wants to be able to maintain connections through handoffs, so that he can transfer long files without disruption while cycling across town. His first idea is to modify the File Transfer Protocol to allow handoff. Whether or not he figures out how to do the modification, what is the main defect in this proposal?

e. ***Briefly*** suggest a way of accomplishing handoff entirely within the network layer.

Quiz #2 1995

1. *(10 points)* Consider the UNIX file system naming hierarchy illustrated below:



You have been handed the following path name:

/projects/6.033/quiz2/prob1

and you are about to resolve the third component of that path name, the name "quiz2".

1a. Circle the closure, and write the word "closure" next to the thing you have circled.

1b. Circle the context, and write the word "context" next to the thing you have circled.

2. *(25 points)* To obtain high throughput, communication applications pipeline messages: after sending the first packet of a message, they immediately send the next one without waiting for an acknowledgement of the first packet. A key question is how to establish how many packets to pipeline between a request and its acknowledgement (i.e., what is the window size).

(5 points) Assume the client and the server are connected by an 800,000 byte/sec link. Also assume that the client and the server produce and consume packets at the same rate. Using the server's acknowledgements, the client measures the round-trip between itself and the server to be 10 milliseconds. If the client is sending 1000-byte packets, what is the smallest window size that allows the client to achieve 800,000 byte/sec throughput?

2a. *(5 points)* A popular scheme for establishing the window size in the Internet is *slow start*. In a simple version of slow start, a client starts with a small window size and increases the window size. Assume that the client starts with a window size of one packet. Also assume that for every packet, the server responds with an acknowledgement telling the client to double the window size. How long would it take for the client to know it has reached the maximum throughput?

2b. *(5 points)* Another scheme for establishing the window size is *fast start*. In fast start, a client starts by assuming a default large window size (say 8 packets) and sends the server a burst of that size. On the server's response, the window is adjusted to the size that is supplied by the server. If a client starts with a window size of 8, how long does it take for the client to know it has achieved the maximum throughput?

2c. *(10 points)* In the circumstances described above, fast start seems to be preferable. In what circumstances would it be preferable to use slow start over fast start? You may assume that one of the back-pressure techniques used in the Internet is dropping packets.

3. *(25 points)* Secure Inc. is developing an improved version of AFS, Secure AFS (SAFS), which automatically encrypts files to guarantee better privacy of information. When a request to store a file arrives, SAFS enciphers the file using the client's key. On arrival of a request to read a file, SAFS looks up the client key, deciphers the file, and sends the file back to the client. SAFS keeps for each client a separate key to encipher and decipher files.

3a. *(5 points)* The designers of Secure Inc. are wondering how long it would take to crack an enciphered file that is encrypted using RSA with 512-bit key, which is what is typically used today. To crack an RSA-enciphered file one has to factor the key. The designers know from the literature that factoring a 100 decimal digit number takes about 1 month using idle cycles from 300 3-MIPS SUN-3 workstations. It is estimated that factoring an additional 3 decimal digits roughly doubles the computation time needed. How many 3-MIPS computers are needed to factor a 155 decimal digit number (which corresponds to about 512 bits) in one month?

(5 points) If processors are doubling computation performance per year, how many computers would it take to factor a 512-bit key in one month in the year 2000?

3b. *(5 points)* Let's assume that enciphering and deciphering can be done at 250 Kbyte/sec. How much would the throughput be reduced for reading files stored by SAFS, if the current maximum throughput without enciphering/deciphering is 800 Kbyte/sec? (You may assume that enciphering/deciphering cannot be pipelined with sending/receiving.)

3c. *(10 points)* Secure Inc. is also considering adding automatic compression of files to SAFS. Compression reduces redundancy of information in a file so that the file takes less disk space. Should they first compress files, then encipher them, or should they first encipher files and then compress them? Explain your answer briefly.

4. *(15 points)* Baybank is struggling to convince itself of the validity of a message it just received, and has asked your help in what to do next. So far, they know the following two facts to be true:

Louis says (Ben says (Transfer $1,000,000 to Alyssa))
Jim speaks for Ben

According to Ben's account representive, Ben's account has enough money for such a transaction, so if they can convince themselves that Ben really said that, they will do the transfer.

4a. *(5 points)* Which of the following things should they attempt to establish the truth of?

a. Louis speaks for Jim

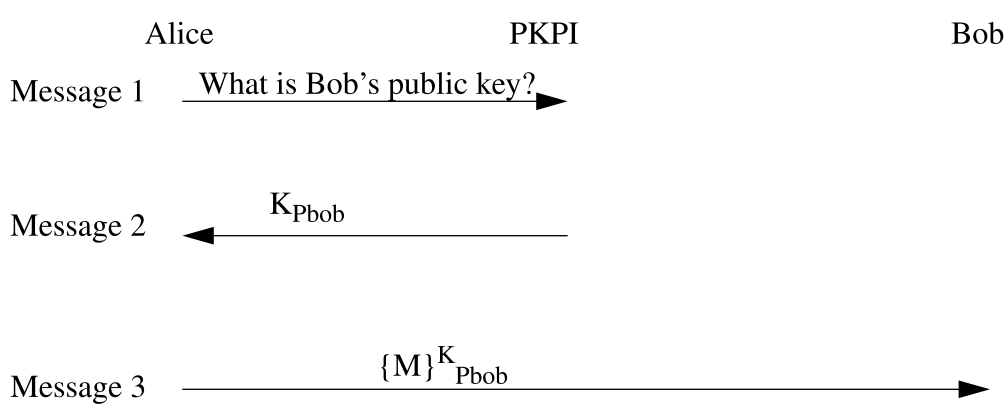
b. Ben speaks for Louis

c. Ben says (Jim speaks for Louis)

4b. *(10 points)* Why?

5. *(25 points)* Ben Bitdiddle and Louis Reasoner have founded a startup company, named Public Key Publication, Inc. (PKPI), whose business is distributing public keys. Their idea is that people who have a key-pair for use with a public-key system need a way of letting other people know the public key of their key-pair. Ben and Louis are not interested in creating keys, but just in acting as a public key distributor.

Ben and Louis have designed the following protocol, in which Alice sends a private message to Bob. They need your help in debugging the protocol. K_{Pxyz} is the public key of principal xyz.



Messages 1 and 2 constitute the PKPI protocol; message 3 is the beginning of Alice's protocol with Bob and is not under the control of PKPI; message 3 is shown here only to place the PKPI protocol in context.

5a. *(5 points)* Louis believes that Eve, the passive eavesdropper, will find that she cannot learn anything by overhearing the PKPI protocol in use. Give an argument that supports Louis' position, or an example demonstrating that Louis is mistaken.

5b. *(10 points)* Louis originally hoped that Lucifer, the active attacker, wouldn't be able to cause any problems, either, but since hearing the 6.033 lecture on the topic he is not sure. Give an example of an active attack that demonstrates that Louis needs to revise the PKPI protocol to protect against Lucifer.

5c. *(10 points)* Ben suggests that the protocol could be improved by changing Message 2. What should Message 2 contain, and how should it be enciphered, so that Alice can be confident that no one but Bob can decipher message 3?

Quiz #2 1996

1. Ben Bitdiddle has been thinking about RPC. He remembers that one of the problems with RPC is the difficulty of passing pointers: since pointers are really just addresses, if the server dereferences a client pointer, it'll get some value from *its* address space, not what the client intended. Ben hopes to get some insight by applying his newfound 6.033 naming knowledge to the problem.

1a. *(15 points)* Match up each concept on the left with the corresponding naming concept on the right. You won't use some of the naming concepts.

____ address space	(1) implicit context
____ pointer	(2) object
____ memory value	(3) multiple contexts
____ pair of address space and pointer	(4) closure
____ dereferencing a pointer in the current address space	(5) search path
	(6) context
	(7) name

Inspired by his naming insights, Ben decides to pass closures instead of bare pointers in his RPC system. Louis Reasoner, excited by Ben's discovery, decides to change *all* end-to-end protocols used on MIT's networks along the same lines.

1b. *(5 points)* Argue for or against Louis's changes, using 6.033 arguments.

2. In a mobile computing environment, locating a user in the face of user movement is an important problem. Consider a system in which each user is assigned a permanent home location, which can be found simply by looking the user up in a name service.

   Alyssa P. Protocol-Hacker is designing an end-to-end protocol for locating these mobile hosts. Her first protocol is very simple: every time a user moves, store a forwarding pointer at the old host, pointing to the new host. This creates a chain of forwarding pointers with the permanent home location at the beginning and the mobile host at the end. Messages meant for the mobile host are sent to the home location, which forwards them along the chain until they reach the mobile host itself. (The chain is truncated when a mobile host returns to a previously visited location.)

Alyssa notices that performance generally gets worse each time she moves her mobile host! "This must be because of the long chains of forwarding pointers," she muses.

Help Alyssa design a more efficient protocol which avoids long chains. Assume that the end-to-end protocol you design is built on top of a network protocol which can reorder messages, but doesn't lose or duplicate them.

(8 points) Alyssa's first try works like this. Each time a mobile host moves, it sends a message to its home location indicating its new location. The home location maintains a pointer to the new location. With this protocol, there are no chains at all. Nodes other than the home node do not maintain forwarding information.

When this protocol is implemented, Alyssa notices that messages regularly get lost when she moves from one location to another. Explain briefly why or give an example.

2a. *(8 points)* Alyssa is disappointed with her first attempt, and decides to start from scratch. In her new scheme, no forwarding pointers are maintained anywhere, not even at the home node. Say a message destined for a mobile host A arrives at a node N. If N can directly communicate with A, then N sends the message to A, and we're done. Otherwise, N broadcasts a search request for A to all the other fixed nodes in the network. If A is near a different fixed node N′, then N′ responds to the search request. On receiving this response, N forwards the message for A to N′.

Will messages get lost with this protocol, even if A moves before the message gets to N′? Briefly explain your answer.

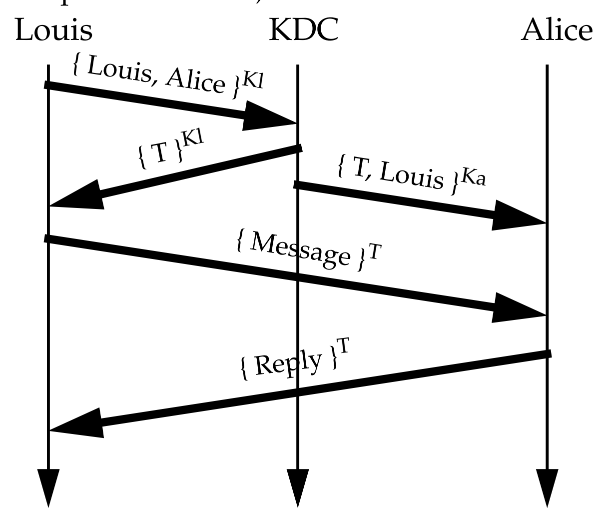
2b. *(8 points)* Unfortunately the network doesn't support broadcast messages efficiently, so Alyssa goes back to the keyboard and tries again. Her third protocol works like this. Each time a mobile host moves, say from N to N′, a forwarding pointer is stored at N pointing to N′. Every so often, the mobile host sends a message to its permanent home node with its current location. Then, the home node propagates a message down the forwarding chain, asking the intermediate nodes to delete their forwarding state. Can Alyssa ever lose messages with this protocol? Explain briefly. (Hint: think about the properties of the underlying network.)

2c. *(8 points)* What additional steps can the home node take to ensure that the scheme in question c never loses messages? Describe briefly.

3. Louis B. Unreasonable has decided that the NSA has gotten their hands on PGP and probably made it weak or insecure. He therefore wants to write his own safer and more secure encryption system. Before doing so, he first wants to understand the most popular schemes of today, RSA and DES. RSA is a public-key system where encryption and decryption each take 1,000µs per block (a block is 0.1Kbyte or less). DES is a private-key system with a single key; encrypting or decrypting with DES only costs 10µs per block. (Only consider passive attackers for this problem.)

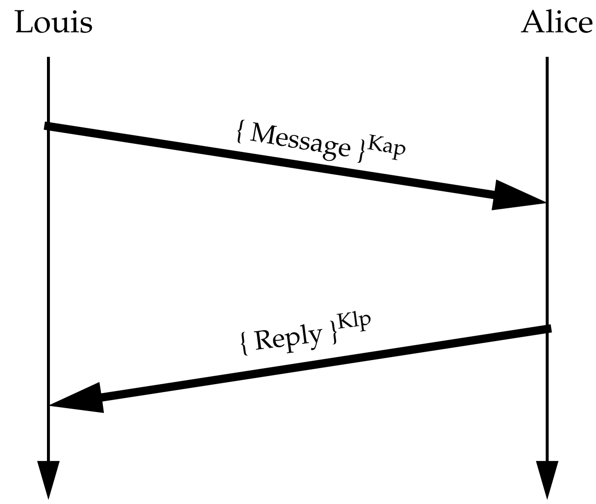
3a. *(6 points)* How long does it take to encrypt a file that is 1,000Kbytes in size, assuming you are using RSA? How long does it take to encrypt that same file using DES?

3b. *(6 points)* Louis was planning on using public-key encryption, but the costs you calculated for him have changed his mind. Now, his system will have central key distribution centers (KDCs) using only private-key encryption. Here's how the protocol works when Louis sends a message to Alice. (Louis's private key is Kl; Alice's private key is Ka; and the temporary private key the KDC sets up for them is T.)



How long will it take for this exchange to complete, that is, from the time Louis starts encrypting the first message to the KDC to the time Louis is done decrypting Alice's reply? *Only consider encryption times, decryption times, and network latency.* Assume the network latency is 2,000µs per message. Assume also that each message fits in one block.

3c. *(6 points)* Alyssa P. Hacker thinks that Louis's reasoning is all wrong, particularly when messages and replies are small. She proposes a public-key encryption system; with her protocol, the same message exchange would look like this. (She assumes that Louis and Alice already know each others' public keys.)

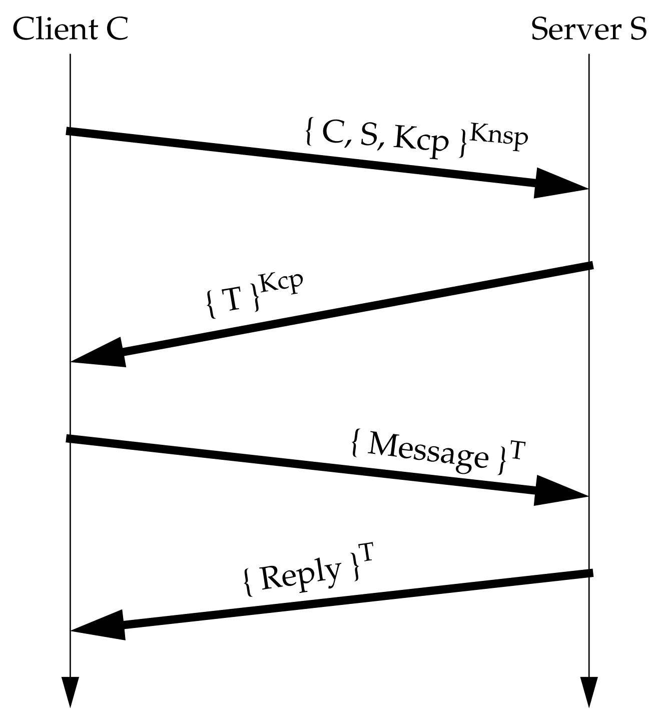


How long will it take for this message exchange to complete with Alyssa's protocol (encryption times, decryption times, and network latencies only)? Again, the message and reply each fit in a block.

3d. *(6 points)* What guarantee does Louis's protocol provide for Alice that Alyssa's protocol doesn't? Assume that Louis keeps T secret.

4. A startup company with an insanely high stock price, the Notsafe Communications Corporation, is designing a protocol for safe and private transactions over the Internet. Rather than rely on centralized security servers like Kerberos, the Notsafe designers are using public-key cryptography to implement their protocol. However, only the initial handshake uses public-key, since the protocol will be used for high-volume data transfers, where public-key cryptography is just too expensive. The handshake will set up a private key for communication between the parties, which then can continue for as much as a day or more.

   The Notsafe protocol has a well-known public key, K_{nsp}, which is used to establish connections. In addition, servers who want to talk the Notsafe protocol must use a Notsafe server program, which includes the secret key, K_{nss}. The protocol is graphically depicted below.



   In the first message, the client sends a handshake including C's public key, encrypted with K_{nsp}: $\{C, S, K_{cp}\}^{Knsp}$. The server decrypts this with K_{nss} and sends a response back to the client including a 32-bit private key T for the rest of the communication, encrypted with K_{cp}. Then, the client decrypts T with K_{cs} and sends messages and receives replies encrypted with T.

4a. *(2 points)* Notsafe's marketing department declares that "the risk of an intruder reading your sensitive data with Notsafe's products is less than one in one trillion." Mrs. Bach O. T. Envelope-Calculation immediately exclaims, "Poppycock!" Given the key that is being used, what might she have been thinking?

4b. *(9 points)* List 3 fundamentally different problems with Notsafe's protocol.

4c. *(5 points)* List one possible problem that could arise from a very poor implementation of the Notsafe server (*not* a fundamental problem with the protocol itself).

4d. *(8 points)* Introduce two changes to the Notsafe protocol which improve its security and explain them briefly. *You are not expected to create a completely secure protocol.*

Quiz #2 1997

1. (*16 points*) Louis Reasoner has been using a simple RPC protocol that works as follows:

```
client -> server:    {nonce, function, arguments}
server -> client:    {nonce, response}
```

The client sets a timer, and if it does not receive a response before the timer goes off, it restarts the protocol from the beginning, repeating this sequence as many times as necessary until a response returns. The server maintains a table of nonces and responses, and when it receives a request containing a duplicate nonce it repeats the response, rather than repeating execution of the function. The client similarly maintains a list of nonces for which no response has yet been received, and it discards any reponses for nonces not in that list, assuming that they are duplicates. One possible response is “unknown function”, meaning that the server received a request it didn’t know how to handle. A lower network layer checksums all packets and discards any that are damaged in transmission.

Louis wants to make this protocol secure against eavesdroppers. Louis has discovered that the client and the server already share a secret key, K_{cs}, for a symmetric-key encryption system. So the first thing he tries is to encrypt the requests and responses of the simple RPC protocol:

client -> server: $\{\text{nonce, function, arguments}\}^{Kcs}$
server -> client: $\{\text{nonce, response}\}^{Kcs}$

This seems to work, but Louis has heard that if you use the same key to repeatedly encipher predictable fields such as function names, someone may eventually discover the key by cryptanalysis. So he wants to use a different key for each RPC call. To minimize the coding effort, he changes the protocol to work as follows:

client -> server: $\{K_{tn}\}^{Kcs}$
client -> server: $\{\text{nonce, function, arguments}\}^{Ktn}$
server -> client: $\{\text{nonce, response}\}^{Ktn}$

in which K_{tn} is a one-time key chosen by the client to be used only for the n’th RPC call before beginning the protocol. When the server receives a key, it deciphers it and uses it until the server gets another key message. Louis figures that since K_{cs} is now being used only to encipher randomly selected keys, it should be safer from cryptanalysis.

At first, this protocol, too, seems to work. Then Louis notices that the client is receiving the response “unknown function” much more often than it used to.

1a. Explain why, using a timing diagram to demonstrate an example of the failure.

1b. Offer a suggestion that should fix this problem.

 Name ______________________

2. *(32 points)* ByteStream Inc. sells three data-transfer products: Send-and-wait, Blast, and Flow-control. Mike R. Kernel is deciding which product to use for his advanced kernel, and he performs a series of back-of-the-envelope calculations to select the appropriate product. Since each product transfers lots of data, the stream has to be broken up into packets. The protocols work as follows:

   Send-and-wait sends one packet and then waits for an acknowledgment before sending the next packet.

   Flow-control uses a sliding window of 8 packets. The sender sends until the window closes (i.e., until there are 8 unacknowledged packets). The receiver sends an acknowledgment as soon as it receives a packet. Each acknowledgment opens the sender's window with one packet.

   Blast uses only one acknowledgment. The sender blasts all the packets in a message to the receiver. The last packet of the blast contains a bit indicating that it is the last packet of the message. After sending all packets in a single blast, the sender waits for one acknowledgment from the receiver. The receiver sends an acknowledgment as soon as it receives the last packet.

   You are asked to compute for each the protocol the maximum throughput. Mike is planning to use a 1,000,000 byte/sec network with a packet size of 1,000 bytes. The propagation time from the sender to the receiver is 500 microseconds. Mike assumes (1) there is no processing time at the sender and the receiver; (2) that the time to send an acknowledgment is just the propagation time ($N = 0$); and (3) the data packets are always 1,000 bytes. He also assumes that the underlying communication medium is perfect (packets are not lost, packets are not duplicated etc.) and that the receiver has unlimited buffering.

   2a. What is the maximum throughput for the Send-and-wait? *Check one.*

   ___1,000,000 byte/sec

   ___666,000 byte/sec

   ___500,000 byte/sec

   ___200,000 byte/sec

   2b. What is the maximum throughput for Flow-control? *Check one.*

   ___1,000,000 byte/sec

   ___666,000 byte/sec

   ___500,000 byte/sec

   ___200,000 byte/sec

   2c. What is the maximum throughput for Blast? *Check one.*

   ___1,000,000 byte/sec

   ___666,000 byte/sec

   ___500,000 byte/sec

   ___200,000 byte/sec

Mike needs to choose one for an application which periodically sends arbitrary-sized messages. He has a reliable network, but his application involves unpredictable computation times at both the sender and the receiver. There is also a 20,000-byte buffer for receiving packets at the receiver.

2d. Which product should he choose for maximum reliable operation? *Check the best answer of the ones listed (i.e., the most specific answer).*

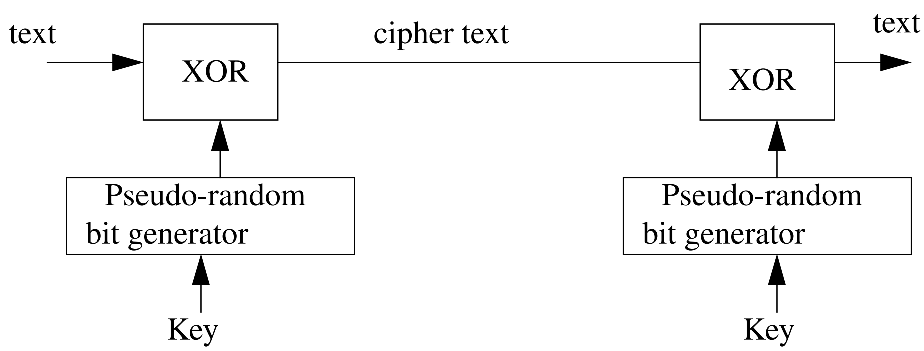
___Send-and-wait, the other might hang.

___Blast, which outperforms the others.

___Flow-control, since Blast will be unreliable and Send-and-wait is slower.

___I cannot tell from the info given (*briefly explain*!).

3. *(24 points)* Because of the recent interest in security in the popular press, ByteStream Inc. decides to extend the functionality of its products with encryption. ByteStream decides to use a simple symmetric-key system:



Both for encryption and decryption ByteStream uses the exclusive-OR (XOR) function as shown above. The pseudo-random number generator generates a stream of random bits. If seeded with the same key, it will generate the same stream of random bits. Messages are encrypted by computing the XOR of the message and the random bits produced by the generator, seeded with some key. The resulting ciphertext is decrypted by computing the XOR of the ciphertext and the random bits produced by the random-bit generator, seeded with some key. If the keys used in encrypting and decrypting are the same, then the resulting decrypted ciphertext will produce the original message. The code for computing the XOR and the pseudo-random bits is publically available, but the key is secret.

To check the implementation, ByteStream Inc. hires a tiger team that include Eve S. Dropper and Lucy Fer. The tiger team verifies that the code for computing the XOR and the random-bit generator does not contain cryptographic weaknesses. The tiger team subsequently studies the following scenario. Alice shares a 200-bit key K with Bob. Alice encrypts a message with K as the seed for the random-bit generator. She sends the resulting ciphertext to Bob. Bob decrypts this message with K as the seed for the random-bit generator. The result after decryption is Alice's message. Assume that every message is equally likely (i.e., Alice's message does not contain redundancy).

3a. Given that Eve sees only the cipher text, is she able to decode Alice's message? *Check the best answer possible of the ones listed (i.e., the most specific answer).*

___ No, since only Alice and Bob know the key, and the pseudo-random generator generates a 0 or 1 with equal probability, Eve has no way of telling what the content of Alice's message is.

___Yes, since with a supercomputer Eve could try out all possible combinations of 0s and 1s for K and check whether they match the cipher text.

___No, since it is hard to compute the XOR of two bit streams.

___Yes, since XOR is a very simple function, and Eve can just compute the inverse of XOR.

3b. Alice and Bob switch to a new shared key K. Lucy mounts an active attack by tricking Alice into sending a message that begins with 500 1s, followed by Alice's original message. Given the ciphtertext can Lucy figure out Alice 's message? *Check the best answer possible of the ones listed (i.e., the most specific one).*

___Yes, since the key is smaller than 500 bits.

___Maybe, but with probability so low that it is negligible.

___No, since only Alice and Bob know the key and the pseudo random generator generates a 0 or 1 with equal probability, Lucy cannot guess Alice's message.

___No, since it is hard to compute the XOR of two bits.

ByteStream is interested in a product that supports two-way communication. ByteStream implements two-way communication by having one stream for requests and one stream for replies. ByteStream seeds both streams with the same key. Since ByteStream worries that using the same key in both directions is a weakness, it asks the tiger team to check the implementation.

The tiger team studies the following scenario. Alice seeds the random-bit generator for the request stream with K and sends Bob a message. Upon receiving Alice's message, Bob seeds the random-bit generator for the reply stream with K, and sends a response to Alice. Again, assume that every request and response is equally likely.

3c. What can Eve deduce about the content of the messages? *Check the best answer possible of the ones listed (i.e., the most specific one).*

___Nothing.

___The content of the request, but not the reply.

___The XOR of the request and the reply, but not the content of either the request or reply.

___The content of both the request and the reply.

4. *(28 points)* ByteStream Inc. extends its encrypted ByteStream products with an authentication protocol. The authentication protocol is based on a key server to simplify key management for users. ByteStream Inc. studies Kerberos, but decides it is too complicated. So, it designs a home-grown version of Kerberos that involves a single centralized ticket service CTS that grants tickets for services and session keys for communication. In ByteStream's system there is no separate ticket-granting service---CTS handles all the requests for tickets and keys.

   CTS maintains a shared key for every principal. The protocol for a client C accessing service S is as follows. C sends a request to CTS, in the clear, containing C's name, S's name, and a nonce N (which is unique over time and unpredictable):

   Message 1: C $\longrightarrow$ CTS : {C, S, N}

   CTS sends a temporary session key K, S's name, C's nonce from Message 1, and a ticket for S to C, all encrypted using the shared key for C. The ticket consists of the session key, C's name, and CTS's current local time T, all encrypted using the shared key of S, K_S:

   Message 2: CTS $\longrightarrow$ C : $\{K, S, N, Tick\}^{KC}$, where Tick = $\{K, C, T\}^{KS}$

   C accepts Message 2 provided that S's name and N match up with what it sent to CTS in Message 1. C initiates communication with S by sending the ticket in the clear, together with an authenticator containing the new current time T' and a key K':

   Message 3: C $\longrightarrow$ S : $\{\{T', K'\}^{K}, Tick\}$

   S decrypts Message 3, and accepts it if T (inside Tick) is no more than an hour old and T' is no more than five minutes old. Once it has accepted the message, S is prepared to communicate with C using key K' for encryption of all communications, until T becomes stale (older than an hour).

   The following questions ask you to describe attacks if various mechanisms used in the ByteStream authentication protocol are left out. You might consider active and passive attacks such as eavesdropping, spoofing, and replay, among others.

   4a. If the encryption with K_C of Message 2 were omitted what attacks are possible? Give one example scenario if one exists.

   4b. If T and T' are omitted from the protocol what attacks are possible? Give one example scenario if one exists.

   4c. If S is omitted from Message 2 what attacks are possible? Give one example scenario if one exists.

Quiz #2 1998

I. Any of this sound familiar?

1. The best example of an *end-to-end argument*, as Saltzer *et al.* uses the term, is:

   A. If you laid all the web hackers in the world end to end, they would reach from Cambridge to CERN.

   B. Every byte going into the write end of a UNIX pipe eventually emerges from the pipe's read end.

   C. Even if a chain manufacturer tests each link before assembly, he'd better test the completed chain.

   D. Per-packet checksums must be augmented by a parity bit for each byte.

   E. All important network communication functions should be moved to the application layer.

(circle letter to left of best answer)

1. [6 points]: Ethernet cards have unique Ethernet addresses built into them. What role do these play in an IP network?

   A. None. They are there for Macintosh compatibility only.

   B. A portion of the Ethernet address is used as the IP address of the computer using the card.

   C. They provide routing information for packets destined to non-local subnets.

   D. They are used as private keys in the Security Layer of the ISO protocol.

   E. They provide addressing within each subnet for an IP address resolution protocol.

   F. They provide secure identification for warranty service.

(circle letter to left of best answer)

2. [6 points]: A UNIX directory is a file comprising a sequence of *directory entries*, each of which contains

   A. The ASCII name and logical file number of a file.

   B. A list of physical disk blocks occupied by a file.

   C. Permission information for a file for the owner, for members of the owner's group, and for all users.

D. Time and date of last modification of a file.

E. All of the above.

F. None of the above.

(circle letter to left of best answer)

3. [6 points]: To client software, a notable difference between Birrell's RPC and ordinary local procedure calls is:

A. None. That's the whole point of RPC!

B. There may be multiple returns from one RPC call.

C. There may be multiple calls for one RPC return.

D. Recursion doesn't work in RPC.

E. The runtime system may raise a new type of exception as a result of an RPC.

F. Arguments to RPCs must be scalars.

(circle letter to left of best answer)

4. [6 points]: When an Ethernet station detects interference while transmitting, it

A. Completes the transmission of the packet normally.

B. Immediately stops transmitting, and retransmits the packet later.

C. Records the address of the interfering station, and sends it a subsequent *interference notification* packet.

D. Jams the Ethernet briefly.

E. Sends Email to Janet Reno.

F. Sets the *possibly corrupt* bit at the very end of the packet.

G. None of the above.

(circle letter to left of best answer)

5. [6 points]: Which, if any, of the following “flaws” in UNIX software were exploited by the internet worm? **Circle all answers that apply:**

   A. Unchecked array bounds allowed buffer overruns.

   B. Unsafe debugging commands were available to untrusted traffic.

   C. At many sites the password file was still encrypted with the default key.

   D. “Trusted” machines could bypass authorization protocols.

   E. Unencrypted passwords were sent over networks where malicious software could capture them.

(circle letter to left of ALL THAT APPLY)

 Name ______________________

II. JailNet

The Computer Crimes Correction Facility, a federal prison for perpetrators of information-related crimes, has observed curious behavior among their inmates. Prisoners have discovered that they can broadcast arbitrary binary strings to each other by banging cell bars with either the tops or bottoms of their tin cups, making distinct sounds for "0" and "1". Since such sounds made in any cell can typically be heard in every other cell, they have devised an Ethernet-like scheme for communicating varying-length packets among themselves.

The basic communication scheme was devised by Annette Laire, a CCCF lifer convicted of illegal exportation of restricted information when the GIF she emailed to her cousin in El Salvador was found to have some bits in common with a competent encryption algorithm.

Annette defined the basic communication primitive

```
Send(char *message, char *from, char *to)
{
  Bang(AllOnes);            /* Start with a byte of 8 1's */
  Bang(to);                 /* destination inmate number */
  Bang(from);               /* source inmate number */
  Bang(message);            /* the message data */
  Bang(Checksum(to,         /* Checksum of whole message */
            from, message));
}
```

where the operation **`Bang(data)`** is executed by banging one's tin cup to signal the sequence of bits corresponding to the specified null-terminated character string, including the zero byte at its end. The special string **`AllOnes`** sent initially has a single byte of (eight) 1 bits (followed by the terminating null byte). The high-order bit of each 8-bit character (in **`to`**, **`from`**, **`message`**, and **`Checksum`**) is zero.

Annette specified that the **`to`** and **`from`** strings be the unique numbers printed on every inmate's uniform, since all of the nerd-inmates quickly learn the numbers of each of their colleagues. Each inmate listens more or less continuously for packets addressed to him, ignoring those whose **`to`** field don't match his number or whose checksums are invalid.

6. [11 points]: What function(s) are served by sending the initial byte of all 1s? **Circle all answers that apply:**

   A. Bit framing.

   B. Byte (character) framing.

   C. Packet framing.

   D. Packet Reassembly.

   E. None of the above.

(circle letter to left of ALL THAT APPLY)

 Name ____________________________

Typical higher-level protocols involve sequences of packets exchanged between inmates, e.g.:

> **Annette**: `Send`("I thought the lobster bisque was good tonight", `Annette`, `Ty`);
>
> **Ty**: `Send`("Yes, but the filet was a bit underdone for my tastes", `Ty`, `Annette`);

where the symbols `Annette` and `Ty` are bound to character strings containing the uniform numbers of Annette and Ty, respectively.

Of course, prison guards quickly catch on to the communication scheme, and even inject messages of their own:

> **Guard**: `Send`("Oh yeah? Then it's ALPO for you tomorrow!",
> `JimmieTheGeek`, `Annette`);

Such experiences motivate Ty Debole, the inmate in charge of cleaning, to add security measures to the JailNet protocols. Ty reads up on public-key cryptography and decides to use it as the basis for JailNet security. He chooses a public-key algorithm and asks each inmate to generate a public/private key pair and tell him the public key.

- `Key` represents the inmate's public key. Since Ty runs the CCCF laundry, he prints the numbers on inmate's uniforms. He replaces each inmate's assigned number with a representation of `Key`;
- `$Key` is the inmate's secret key. This key is known only to the inmate whose uniform bears `Key`.

Ty assures each inmate that so long as they don't reveal their secret `$Key`, nobody else—inmates or guards—will be able to determine it. Inmates continue to address each other by the numbers on their uniforms, which now specify their public `Key`s.

7. [6 points]: What is an assumption on which Ty bases the security of the secret `$Key`?

A. `$Key` is theoretically impossible to compute from `Key`.

B. `$Key` takes an intractibly long time to compute from `Key.`

C. `$Key` takes at least as long to compute from `Key` as the generation of the `Key`, `$Key` pair.

D. There is a reasonably efficient way to compute `$Key`, but it's not generally known by guards and inmates.

E. None of the above.

(circle letter to left of best answer)

Ty then teaches inmates 4 cryptographic algorithms for messages of up to 1,500 bytes.

- **`Encrypt(message, Key)`** returns a cyphertext string
- **`Decrypt(cyphertext, $Key)`** returns a message string
- **`Sign(message, $Key)`** returns a signature string
- **`Verify(message, Key, signature)`** returns true or false

These functions have the following properties true for every choice of **`Key`** and **`message`**:

- All 4 functions are efficiently computable given the appropriate arguments.
- It is computationally infeasible to derive any information about **`message`** from **`Encrypt(message, Key)`** without knowing **`$Key`**.
- Without **`$Key`**, it is computationally infeasible to create a **`signature`** such that there exists a **`message`** on which **`Verify(message, Key, signature) = true`**.
- **`Decrypt(Encrypt(message, Key), $Key)`** = **`message`**.
- **`Verify(message, Key, signature)`** verifies a digital signature. It returns true if and only if **`signature`** is a result of **`Sign(message, $Key)`**; otherwise it returns false.

Ty proposes improving the security of communications by replacing calls to Send by calls like

```
Send(TyCode(message, from, to), from, to);
```

where **`TyCode`** is defined as

```
char *TyCode(char *message, char *from, char *to)
   {  return Encrypt(message, to); }
```

Ty and Annette are smugly confident that although the guards might hear their conversation, they won't be able to understand it since the encrypted message appears as gibberish until properly decoded.

The first use of **`TyCode`** involves the following message, received by Annette:

`Send`(**`TyCode`**("Meet me by the wall at ten for the escape",**`Ty`**, **`Annette`**),
`Ty`, **`Annette`**);

8. [6 points]: What computation did Annette perform to decode Ty's message? Assume **`rmessage`** is the message as received, **`message`** is to be the decoded plaintext, and that **`$Annette`** and **`$Ty`** contain the secret keys of Annette and Ty, respectively.

   A. **`message = Verify(rmessage, Ty, $Annette);`**

   B. **`message = Encrypt(rmessage, $Ty);`**

   C. **`message = Encrypt(rmessage, Ty);`**

   D. **`message = Decrypt(rmessage, $Annette);`**

   E. **`message = Sign(rmessage, $Ty);`**

   F. **`message = Decrypt(rmessage, Annette)`**;

   G. None of the above

(circle letter to left of best answer)

After receiving the message, Annette sneaks out at ten to meet Ty who she expects will help her climb over the prison wall. Unfortunately Ty never shows up, and Annette gets caught by a giggling guard and is punished severely (early bed, no dessert). When she talks to Ty the next day, she learns that he never sent the message; she expects that it was sent by a guard, but is puzzled since the cryptography is secure.

9. [6 points]: What is the most likely explanation?

   A. Annette's **`$Key`** was compromised during a search of her cell.

   B. Some other message Ty sent was garbled in transmission, and accidentally came out "Meet me by the wall at ten for the escape".

   C. Annette's secret key was broken by a dictionary attack.

   D. Ty's secret key was broken by a dictionary attack.

   E. Annette was victimized by a replay attack.

   F. None of the above.

(circle letter to left of best answer)

Annette's friend Cert Defy, on hearing this story, comes up with a new cryptographic function. He defines the procedure

```
char *Cert(char *message, char *A)
  { return Append(message, Sign(message, A)); }
```

Here the **`Append`** operation concatenates its arguments with recognizable separators so that the result can easily be parsed into its original components. Unfortunately, Cert is placed in Solitary Confinement before fully explaining how to use this function, though he did state that sending a message with

```
Send(Cert(message, A), from, to)
```

can assure the receiver of the integrity of the message body and the authenticity of the sender's identity.

10. [6 points]: When Ty sends a message to Annette what value should he supply for **`A`**?

    A. **`Encrypt(Annette, $Ty)`**
    B. **`Ty`**
    C. **`$Ty`**
    D. **`Annette`**
    E. **`$Annette`**
    F. None of the above

(circle letter to left of best answer)

After determining the answer to question 11, Annette receives a packet purportedly from Ty. She splits the received packet into **`message`** and **`signature`**. **`Verify(message, Ty, signature)`** returns true.

11. [11 points]: Which of the following can Annette conclude about **`message`**?

    A. **`message`** was initially sent by Ty.
    B. The packet was sent by Ty.
    C. **`message`** was initially sent to Annette.
    D. Only Annette and Ty know the contents of **`message`**.
    E. If Ty sent **`message`** to Annette and Annette only, then only they know its contents.
    F. **`message`** was not corrupted in transmission.

(circle letter to left of ALL THAT APPLY)

 Name ____________________

Annette, intrigued by Cert's contribution, decides to combine **`Send`**, **`TyCode`**, and **`Cert`** to achieve both authentication and confidentiality. She proposes to use **`NewSend`**, combining both features:

```
NewSend(char *message, char *A, char *from, char *to)
{ Send(TyCode(Cert(message, A), from, to), from, to); }
```

Annette engages in the following conversation:

Ty: **`NewSend`**("Let's escape tonight at ten", **`Ty`**, **`Annette`**);
Annette: **`NewSend`**("Not tonight, Seinfeld is on", **`Annette`**, **`Ty`**);

The following night, Annette again gets the message

Ty: **`NewSend`**("Let's escape tonight at ten", **`Ty`**, **`Annette`**);

Once again Annette goes to meet Ty at ten, but Ty never shows up. Eventually Annette gets bored and returns. Ty subsequently disclaims having sent the message. Again, Annette is puzzled by the failure of her allegedly secure system.

12. [6 points]: What is the most likely explanation?

    A. A guard has discovered a general way to determine **`Key`** from **`$Key`**, and is exploiting this at Annette's expense.
    B. Some other message Ty sent was garbled in transmission, and accidentally came out "Let's escape tonight at ten".
    C. Annette's secret key was broken by a dictionary attack.
    D. Ty's secret key was broken by a dictionary attack.
    E. Annette was victimized by a replay attack.
    F. None of the above.

(circle letter to left of best answer)

Pete O'Fender, who has been in and out of CCCF at regular intervals, wants to extend the security protocols to deal with JailNet key distribution. Whenever he's jailed, Pete is placed directly into Solitary Confinement where he has no contact with inmates (except via bar banging), and where the TV gets only 3 channels. The problem is complicated by the facts that (a) Everyone (including Pete) forgets Pete's uniform number as soon as he leaves, so when he returns he can't just re-use the old key; (b) Pete may not even remember the key for Ty or other trusted long-term inmates; (c) Pete is issued an unnumbered uniform while in Solitary, and (d) guards often pose as newly-jailed Solitary occupants to learn inmate secrets. Pete asks you to devise JailNet key distribution protocols to address these problems.

13. [11 points]: Which of the following are true of the *best* protocol you can devise, given the assumptions stated about Encrypt, Decrypt, Sign, and Verify? **Circle all answers that apply:**

    A. Assuming Pete is thrust into Solitary remembering no keys, he can devise a new **`Key/$Key`** pair and broadcast **`Key`**. Using this **`Key`**, Ty can be assured that messages he sends to Pete are confidential.

    B. Assuming Pete is thrust into Solitary remembering no **`Key`**s, he can't convince inmates that they aren't communicating with a guard.

    C. If Pete remembers Ty's uniform number and trusts Ty, an authenticated broadcast message from Ty could be used to remind Pete of other inmates' uniform numbers without danger of deluding Pete.

    D. Even if Pete remembers a trusted inmate's uniform number, any communication *from* Pete can be understood by guards.

    E. Even if Pete remembers a trusted inmate's uniform number, any communication *to* Pete might have been forged by guards.

III. Final Jeopardy

14. [1 point]: Give your best guess as to your grade on this quiz, *outside of this last problem*. Note that the maximum score on these earlier problems is 99; you get credit for this problem if your guess is within 3 points of your grade on those.

Your estimate of Quiz 2 Score: ___________________