Lab #5

Out : October 16, 1997; 2:30 pm
Due : October 23, 1997, 12:30 pm

Experiment 1:

Although you are raking in the profits with your new online casino, some of your players have begun to suspect you of cheating them. In order to preserve the pristine image of offshore gambling, the Gambling Comission has recommended that you change the method with which players at your Internet Roulette tables encrypt their guesses. Instead of using the addition cypher, as before, you have been instructed to employ a one-way function. The revised scheme works as follows:

- The player picks a number between 0 and 5 as her *guess*.
- Using a publicly-known one-way function $h(x)$, the player computes $h(guess)$ and sends the result to the casino along with a bet.
- The casino "spins the wheel" and sends the player the outcome.
- The player sends the casino her plaintext guess.
- The casino then computes $h(guess)$ for the plaintext guess and verifies that it is equal to the value of $h(guess)$ the player originally sent them.
- The casino then adds the amount of the bet to the player's account if she won, or subtracts it if she lost.

The Comission has given you a one-way function to use: $h(x) = 2^x \pmod{97931}$.

Despite this change of protocol, you are nevertheless determined to maximize your profit. To this end, you will devise a new way of cheating as the casino. Your goal is to make certain that the player consistently loses at the virtual roulette table.

When you open this experiment, you will be presented with a setup consisting of a blank calculator labeled "Casino", a calculator set up to act as a player, and a hand-in transcript. The player calculator will write "Win" or "Lose" to the Hand-in transcript based on the outcome of each turn. You must make certain that the player consistently loses. Submit electronically when the Hand-in transcript contains a record of 20 sequential losses. Please submit on paper a brief description of the method you used to cheat.

In Brief...	
We give you:	A calculator set up to act as a player, a hand-in transcript, and a calculator for you (acting as the casino).
Your task:	Program your calculator to perform the casino's portion of the protocol.
Hand in:	Submit electronically once the player has lost 20 sequential turns, and hand-in on paper a brief description of how you cheated.

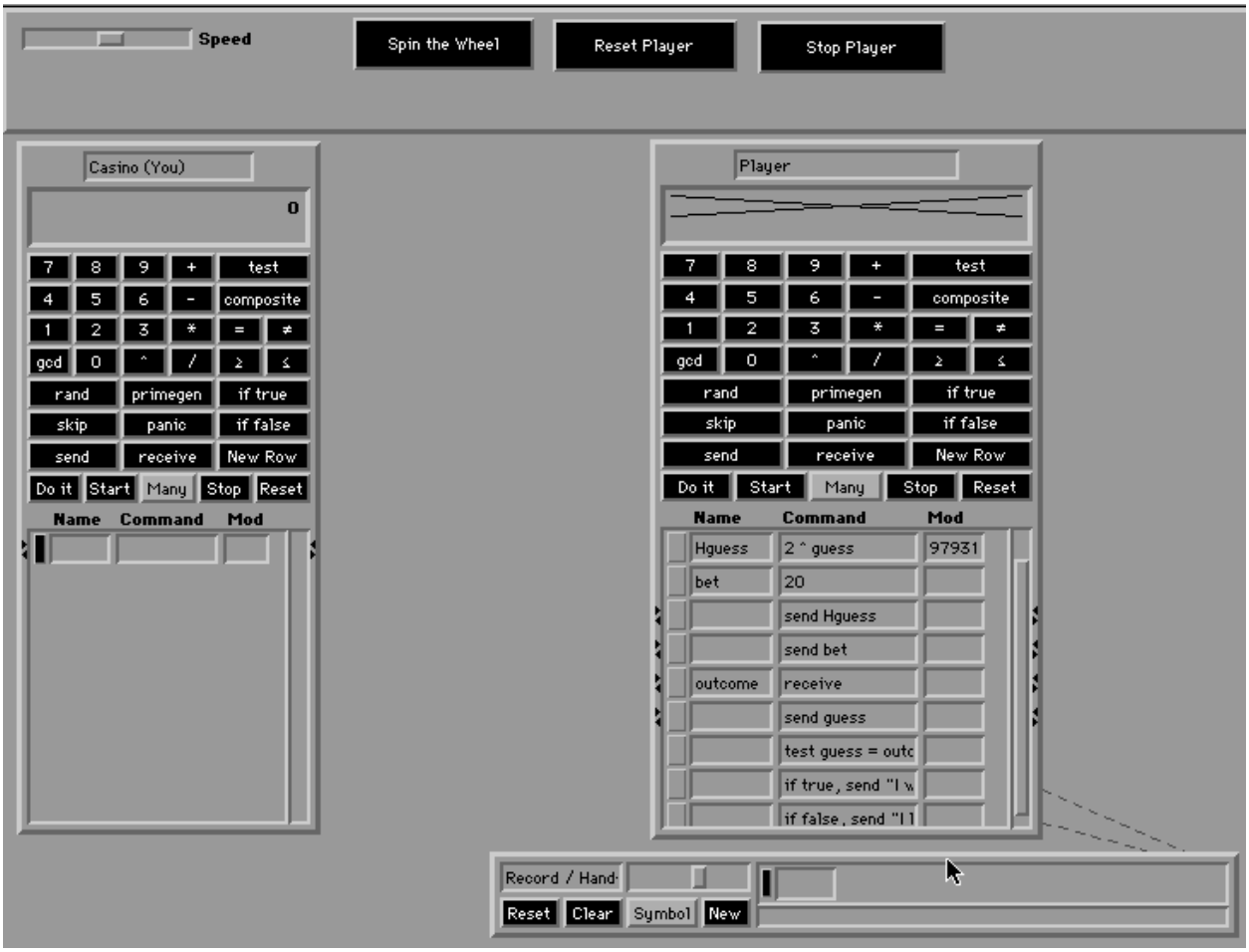

Figure 1: Experiment 1

Experiment 2:

In the next few experiments, you will be working with schemes for logging into a computer using a password. The computer will need to keep track of the passwords associated with each user. This is usually done by storing the passwords in a file. In the MarkCalc, we will represent the password file as a substitution box (which, given a username, outputs the associated password).

(a) In this experiment, you will observe a login session between Alice and a computer. The protocol, in this instance, consists of Alice sending her name and password as plaintext to the computer. The computer then retrieves the password for Alice from the password file and compares it with the password Alice sent. If they match, the login is successful. The potential for Eve to tap the wires from Alice to the computer is pretty obvious. Your method of eavesdropping will not consist of monitoring the lines of communication between Alice and the computer (you will notice that those wires are "secure"). Rather, you will be monitoring access to the password file, which is left insecure.

There is nothing to hand in for this part, you are simply asked to observe and collect information that will help you in part b.

In Brief...	
We give you:	A calculator representing a computer, a calculator representing a user (Alice), a substitution box representing the password file, and a calculator for you (Eve).
Your task:	Observe the protocol.
Hand in:	Nothing to hand in.

(b) After watching the above exchange take place, you will then log into the same computer, posing as Alice.

In Brief...	
We give you:	A calculator representing a computer, a substitution box representing the password file, and a calculator for you.
Your task:	Log into the computer as "Alice".
Hand in:	Submit electronically when the computer displays "Valid Login".

Experiment 3:

As Experiment 2 demonstrates, storing plaintext passwords is dangerous. It doesn't seem that we have much choice in the matter, however, given that the computer must somehow be able to verify that a user is logging in with the correct password. The solution to this problem is to use one-way functions. Instead of containing plaintext passwords, the password file contains the images of the passwords under a one-way function. The computer performs the one-way function on the password sent by the user, and verifies that the output matches the value stored in the password file.

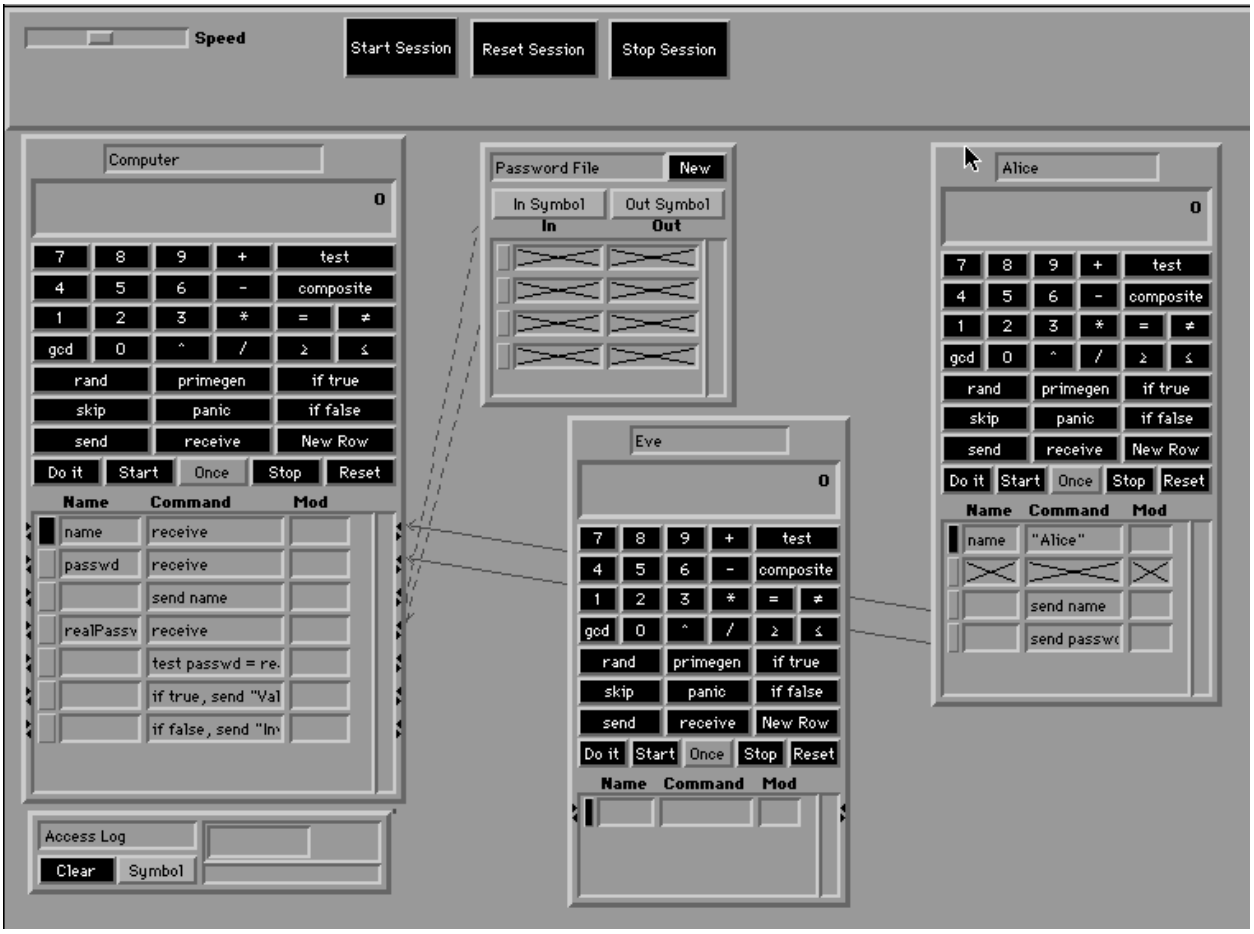

Figure 2: Experiment 2

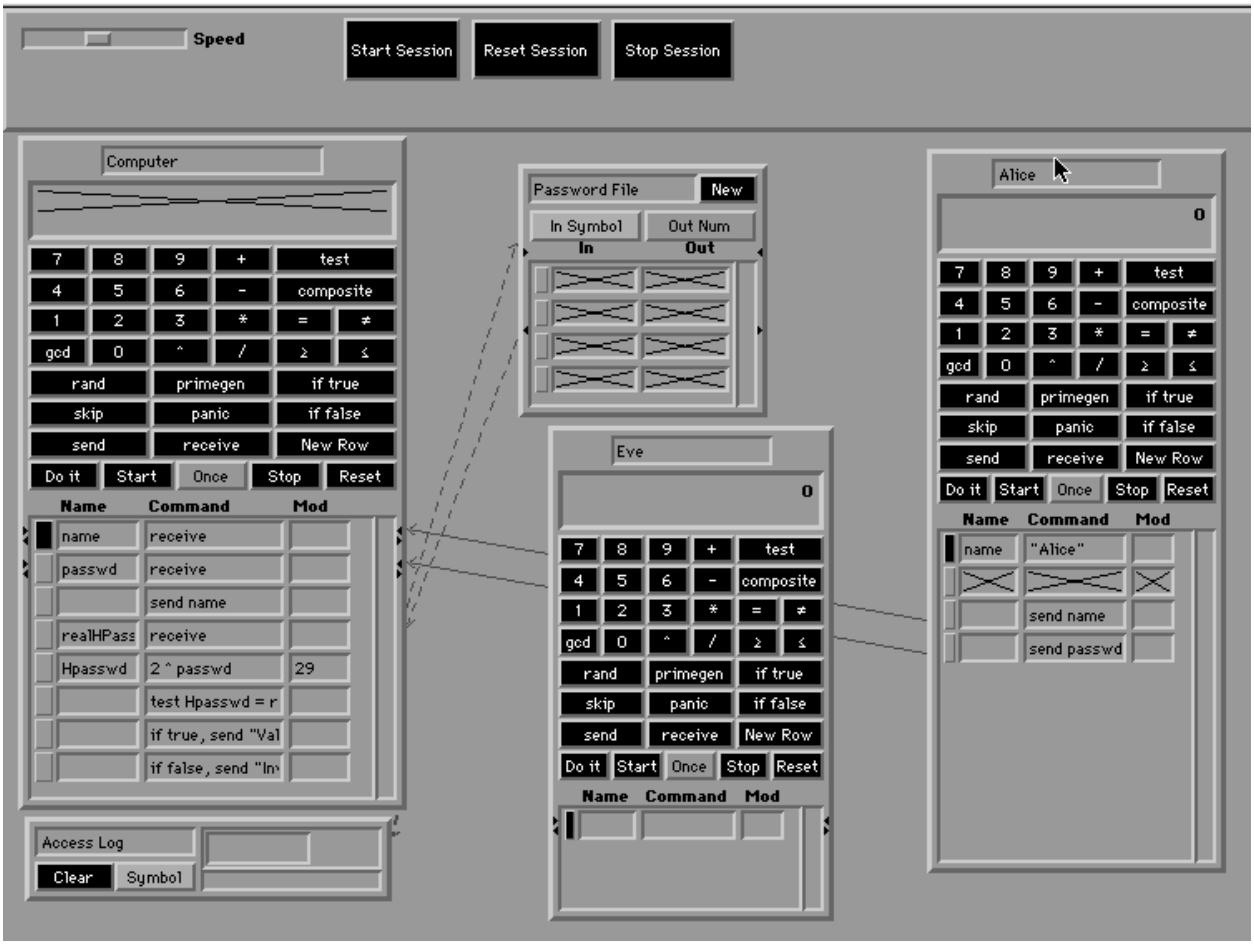

Figure 3: Experiment 3

(a) The designers of the computer used in this experiment are using a purported one-way function: $h(x) = 2^x \pmod{29}$. The astute eavesdropper will notice, however, that 29 is not a very large number. You have already seen how to find the inverse of this operation by hand, in fact. Taking the role of the eavesdropper, you will observe as Alice logs into the computer. As before, you will be snooping the password file to break this scheme.

In Brief...	
We give you:	A calculator representing the computer, a substitution box representing the password file, a calculator representing Alice, and a calculator for you.
Your task:	Observe the exchange.
Hand in:	Nothing to hand in.

(b) After watching this exchange take place, you will then log into the same computer, posing as Alice.

In Brief...	
We give you:	A calculator representing a computer, a substitution box representing the password file, and a calculator for you.
Your task:	Log into the computer as "Alice".
Hand in:	Submit electronically when the computer displays "Valid Login".

Experiment 4:

Having performed the user portion of the login protocol described in Problem 3, you will now take on the role of the computer. You will be using a more secure one-way function: $h(x) = 2^x \pmod{14207}$. In this experiment, you are presented with a password file and a calculator representing a user that wishes to log in. The user calculator, in *many* mode, will read 4 name/password pairs from a transcript and attempt to log in. Some passwords will be valid and some will be invalid. You are responsible for setting up your calculator to check the validity of each login. Each time a user tries to log in, you will send the word "Valid" or "Invalid" to the handin transcript. You may find it useful to put your calculator in *many* mode once you have checked that it works correctly. Submit electronically when you have processed all the login attempts. **Note: You have have been provided with Start, Stop, and Reset buttons at the top of the workspace. Please be sure to press the Reset button to clear the handin transcript if you make a mistake.**

In Brief...	
We give you:	A calculator for you (the computer), a substitution box representing the password file, and a calculator representing users who wish to log in.
Your task:	Authenticate the users.
Hand in:	Submit electronically when you have processed all the logins.

Experiment 5:

In this experiment, you will be timing exponentiation calculations. The goal is to compare two different algorithms for performing this operation. For each part below, you will be given a setup to compute $2^e \pmod{719}$. There will be a transcript in each experiment, labeled "Exponents", which contains several different exponents which you will try in turn.

After timing how long it takes to compute the exponentiations with each algorithm, plot the results on a graph. Use the x-axis to represent the values of the exponents, and the y-axis to represent time. Please make sure to label your axes clearly. Make it clear in your graph which data-points you used to draw the graph (Different students might have different data-points). You will hand these graphs in on paper to the handin bin. There are no electronic submissions for either part.

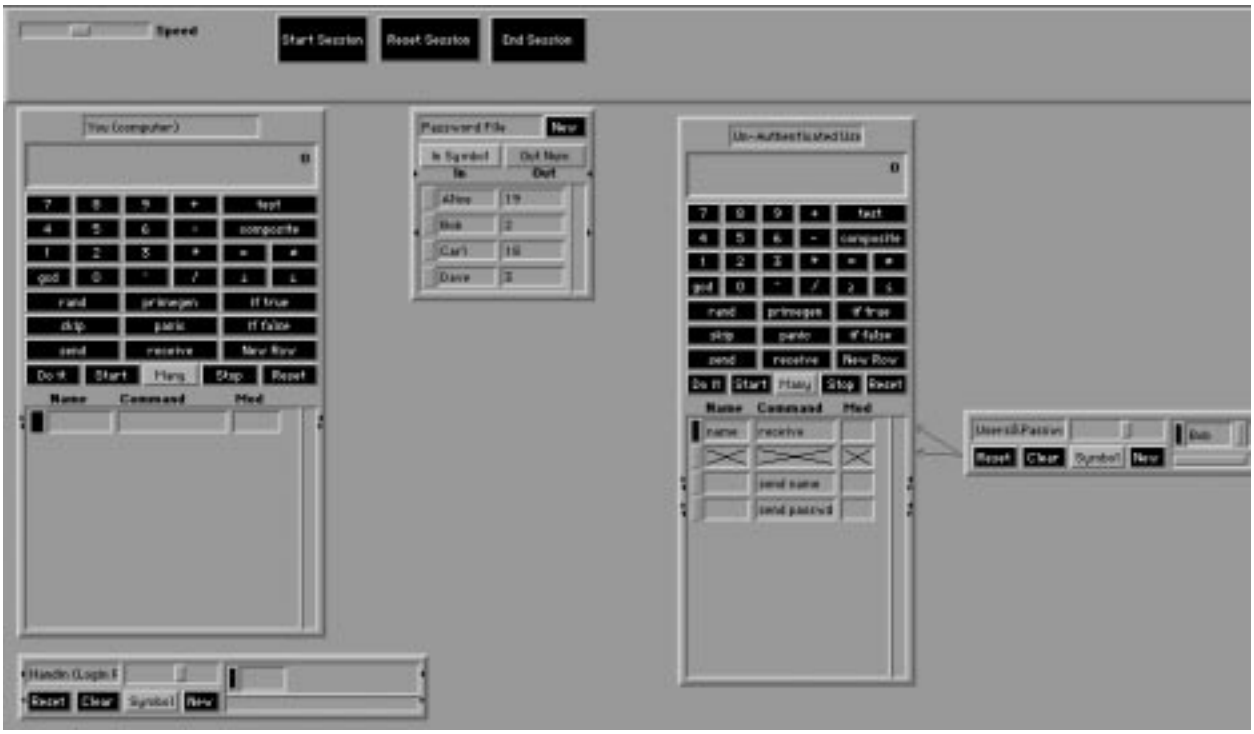

Figure 4: Experiment 4

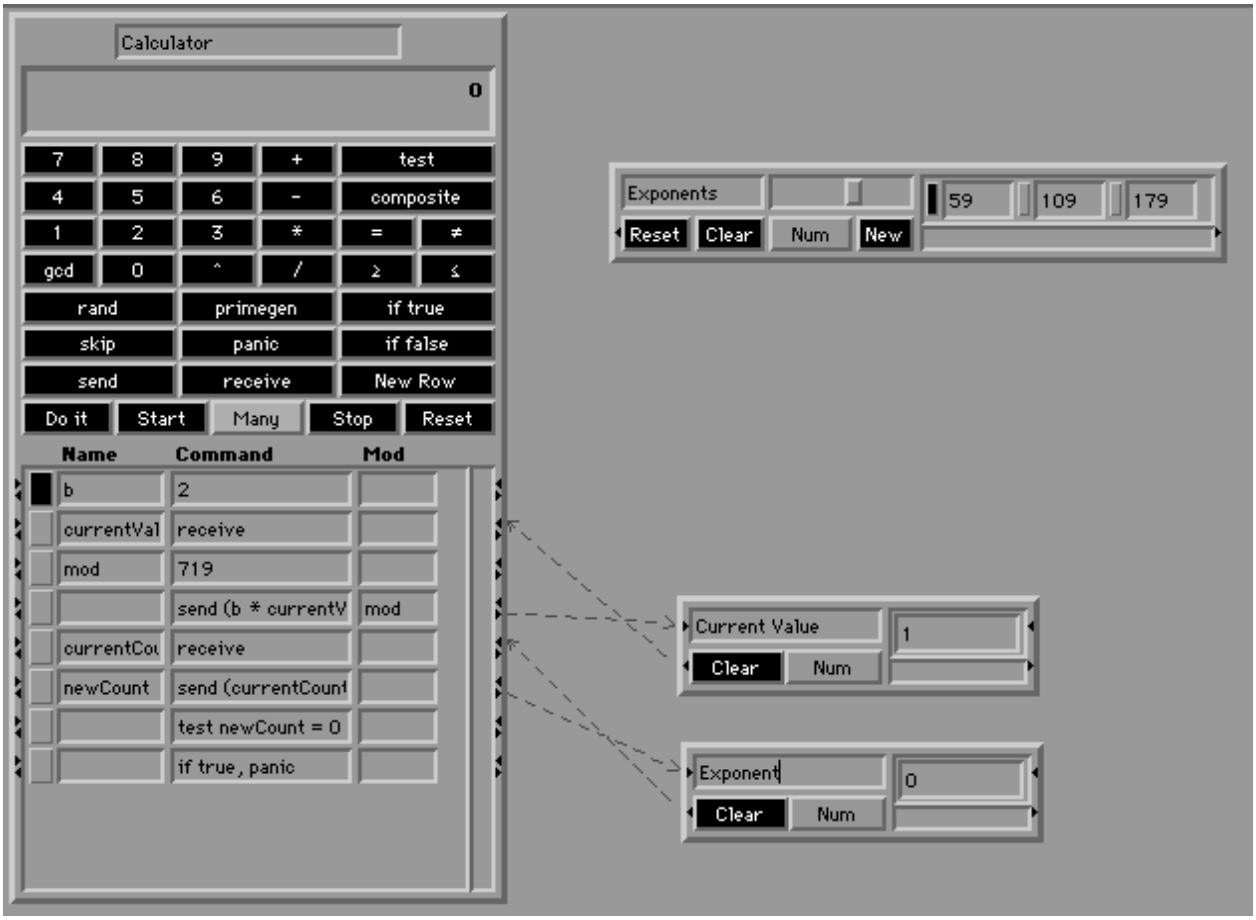

Figure 5: Experiment 5a

(a) The naive algorithm for calculating $2^e \pmod{719}$ involves e separate multiplication operations. In the MarkCalc, this will be performed by a calculator in *many* mode which will read a "current value" from a cell, multiply it by 2, and write the result back to the cell. It will do this e times, keeping track of the number of multiplications performed thus far.

When you open the experiment, you will see a setup consisting of a transcript with various values to use as exponents, a calculator, and a cell for keeping track of the number of multiplications completed. You will need to enter each exponent, in turn, into the cell used to keep track of the number of multiplications (labeled "Exponent") and then run the experiment. The calculator will enter "panic" mode when it is done to let you know that it is finished.

In Brief...	
We give you:	A setup which performs e multiplications in order to exponentiate 2 to the power of e, and a transcript containing various values of e to try.
Your task:	For each exponent e, enter that value into the cell labeled "Exponent" and start the calculator. Time how long it takes for each exponent you have been given.
Hand in:	Submit a graph of your data on paper.

(b) In this experiment, we provide you with a setup which performs a repeated-squaring calculation with the same values as in part a. The algorithm may be a little complicated, but it is not necessary to understand it for this experiment, since you are only concerned with timing its operation. The transcript labeled "Exponents" contains the exponents to use in your experiments. Your task for this part is to time how long it takes the repeated-squaring setup to compute the result. For each exponentiation to be computed, you must perform the following steps to set up the calculator:

- Enter a value of 1 in the cell labeled "Accumulator".
- Enter a value of 2 in the cell labeled "Power".
- Convert the value for the exponent into a binary representation and enter it into the transcript labeled "Exponent" as a series of 1-bit entries, beginning with the least-significant-bit (i.e., the 1's place).

In Brief...	
We give you:	A setup which performs repeated squaring in order to exponentiate 2 to the power of e, and a transcript containing various values of e to try.
Your task:	Time how long this takes for each exponent.
Hand in:	Submit a graph of your data on paper.

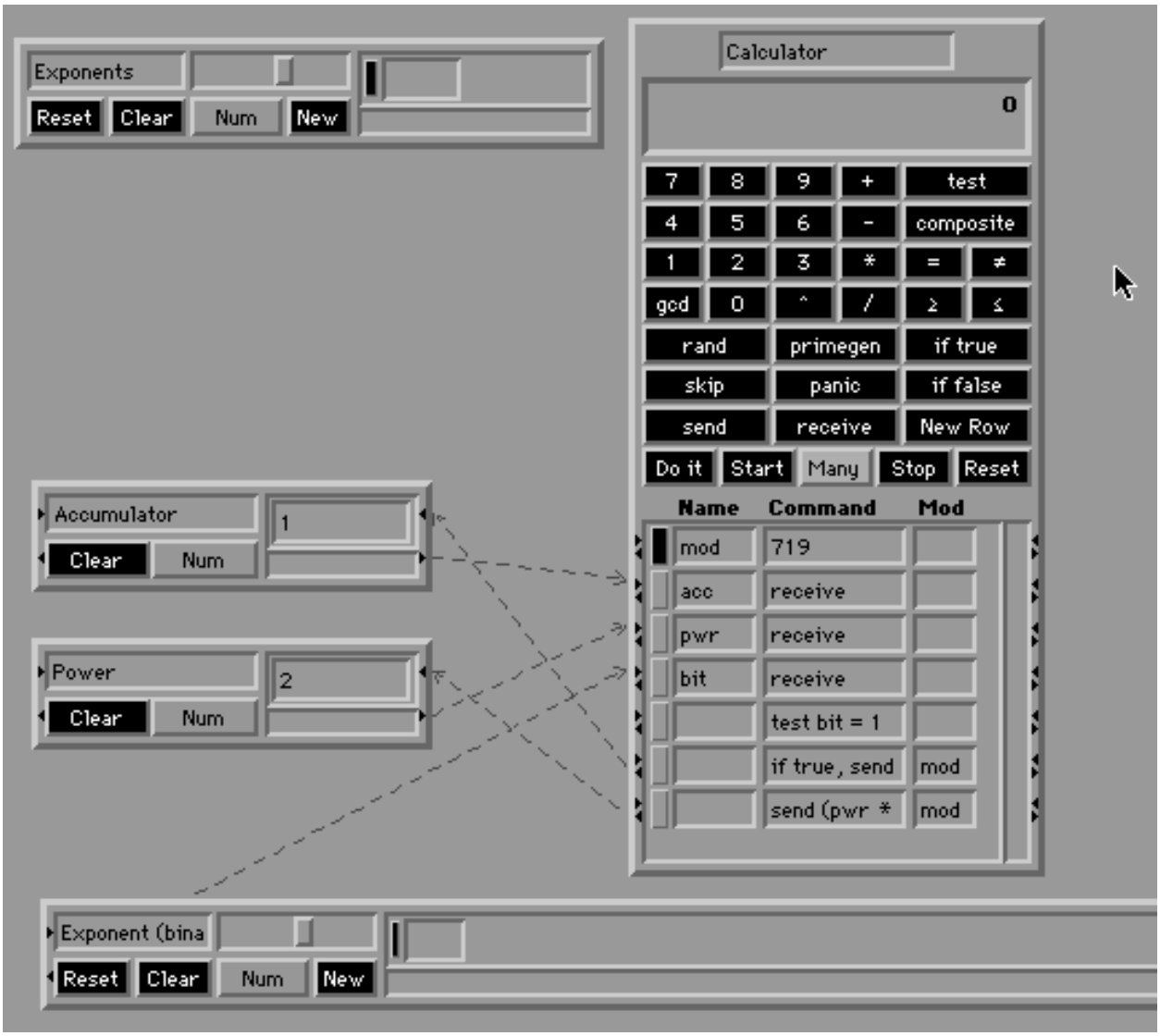

Figure 6: Experiment 5b