Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Score: \_\_\_\_\_ / 20

CPSC 42500 Homework 5: Secure Hashing

1. **(1 point)** What is the square root of 2414 ?

1. **(2 pts)** Say there are 8 people in a room. Each person shakes every other person’s hand once, and nobody shakes the same person’s hand more than once. How many handshakes in total will take place?

1. **(4 pts)** Recall from class that the **parity** function is a very simple hash function producing a one-bit digest, which is 0 if the number of bits in the message is even and 1 if it is odd.   
     
   Consider a communication system that uses 7 bits out of every byte to transmit data, and reserves the 8th bit (the one on the right) to transmit the parity of the other seven bits, for purposes of error checking.   
     
   Below is the binary representation of some bytes that were transmitted using this system. For each one, tell whether there is reason to suspect an error in transmission, giving the reason for your answer.  
     
    10110011  
     
     
    10101100  
     
     
    00110011  
     
     
    11101111
2. **(3 pts)** Explain why a hash function is inappropriate to use for encrypting data.

1. **(2 pts)** Say you are given a hash function that produces a 160-bit digest, along with a message digest *d*. Using random guessing, how long would you expect it to take to find a preimage?

1. **(2 pts)** Say you are given a 160-bit hash function as in the previous question. Using random guessing, roughly how long should it take to find two messages with the same digest?

1. For this question you will do an experiment to carry out your own partial second preimage attack. The site <http://www.xorbin.com/tools/sha256-hash-calculator> has a SHA-256 hash calculator, which allows you to enter any string and see its SHA-256 hash in hexadecimal. As you can verify, the hash of the string “Encryption and authentication” (without the quotes) is the following:  
     
   bab5e003f6ba89f5b99e71763ea4af659f8d3e3002bdc4acc10908e9939fd421  
     
   By random guessing, find another string having a SHA-256 hash that starts with the same hexadecimal digit (b). **Keep track of how many guesses it takes.** Be honest! Count every different string that you try.
   1. **(2 pts)** Write down the string you found that gives a hash starting with b. You don’t need to write its hash.
   2. **(1 point)** How many guesses did it take?
   3. **(3 pts)** How many guesses should it take on average to find a match like this? Did you beat the odds or not?