CSE190A — Final

Yee **Spring '90**

Name and student number (print):________________________________

There are a total of 11 questions on 9 pages. There are 100 points possible.

You may turn in the answers to this exam by electronic mail or by turning in written answers on a printout of this exam. If you plan on the latter, make sure that you have all the pages and write your name on every page. You should look at the class web page periodically; if there are any clarifications that result from emailed questions, the answers will be posted there.

This exam is open notes. You may look at your *own* notes all you want. If you are part of a study group and have shared notes, you may use your *own* copy of notes made by other members of the study group, but you may not obtain new copies of notes. You may *not* use information from the web, except for the class web pages. You may not look at anybody else's notes, exam, or otherwise obtain help from another human being, artificial intelligence, metaphysical entity, or space alien. Please refrain from using any ESP abilities that you may have.

Problem	1	2	3	4	5	6	7	8	9	10	11	**Total**
Score												
Possible	5	10	7	7	10	10	10	10	15	15	1	100

1. [**Expectation Values**] Given a random variable R with the following distribution:

$$R = \begin{cases} 100 & \text{with probability } 0.2 \\ 500 & \text{with probability } 0.3 \\ 1000 & \text{with probability } 0.5 \end{cases}$$

What is the expected value of R?

(5pt)

2. [**Security Economics**] Suppose in your neighborhood of 5,000 dwellings, there were a total of 20 burglaries from homes last year. In two cases, the burglar(s) broke down the door to gain entry to the premises; in 8 cases they gained entry through an unlatched window or door; and in 10 cases they used keys found under door mats, flowerpots, or on top of door frames. The average loss per case was $30,000.

What should the residents in your neighborhood do to improve security? Note that you do not necessary know by how much will any particular security measure lower the odds of being burglarized. (Hint: assume that security measures' "effectiveness" can be measured as the fraction of burglaries that its use would eliminate.)

(10pt)

3. [**Orange Book**] What is a TCB?
(7pt)

4. [**Orange Book**] What is a Trusted Path? What are the required properties?
(7pt)

5. [**Orange Book**] Explain the Bell-Lapadula model? What is the main goal of this model?
(10pt)

6. [**Security Exploits**] Describe the `IFS` security exploit.
(10pt)

7. [**Exploit Classifications**] What does it mean for a security weakness to be a "TOCTOU" problem?
(10pt)

8. [**Differential Fault Analysis**] Explain how Differential Fault Analysis can be used to extract secret cryptographic keys from smartcard-like cryptographic devices. Given the necessary conditions for the attack to succeed.

(10pt)

9. [**Differential Power Analysis**] The `mod_exp` function returns $b^e \pmod{m}$, where b, e, and m are the three parameters to the function.

In the following code, `MAXBITS` is the maximum number of bits in a big integer — all big integers are less than $2^{\texttt{MAXBITS}}$ in size. You may assume that `MAXBITS` is much larger than 32. Assume that the C++ class big_int has the usual arithmetic operators overloaded to perform the expected operations. The `mod_mult(`a`,`b`,`m`)` function provides an efficient way to obtain $a * b \pmod{m}$.

Suppose you had a chip in which the power consumed by the modular multiply operation (`mod_mult`) is independent of the multiplicand and the multiplier values. Suppose further that the value e is secret, m is fixed, and b is under the controll of the user/attacker. Would running the following modular exponentiation algorithm on such a chip be immune to differential power analysis? (If yes, explain why; if no, explain how to extract information via DPA.)

```
big_int mod_exp(big_int b, big_int e, big_int m)
{
      int i;
      big_int y;

      for (i = 0, y = 1; i < MAXBITS && e != 0; i++, x = mod_mult(x,x,m), e = e / 2)
            if ((e % 2) == 1) y = mod_mult(y,x,m);
      return y;
}
```

(15pt)

10. [**Proof of correctness**] Determine the necessary pre-conditions for the following modular exponentiation algorithm `mod_exp2`. Here, `MAXBITS` is the maximum number of bits in a big integer — all big integers are less than $2^{\texttt{MAXBITS}}$ in size. You may assume that `MAXBITS` is much larger than 32.

The function `mod_exp2` returns $b^e \pmod{m}$, where b and m are the big_int parameters, and e is encoded as follows: `exp` is an array of `MAXBITS` elements from $\{0, 1\}$, and the exponent e is given by $e = \sum_{i=0}^{\mathrm{MAXBITS}-1} \texttt{exp}[i]2^i$.

Note that the `mod_exp2` algorithm is different from the `mod_exp` algorithm in the previous problem.

```
big_int mod_exp2(big_int b, char exp[], big_int m)
{
      int i;
      big_int y;

      for (i = MAXBITS, y = 1; --i >= 0; ) {
            y = mod_mult(y,y,m);
            if (exp[i] != 0) y = mod_mult(y,b,m);
      }
      return y;
}
```

(15pt)

11. What is your favorite color? Give its RGB value.
(1pt)