**Reading for Lecture 6:**

1. *D is for Digital:* Part I preface and Chapter 1: What’s in a Computer? pp. 7-19.

2. Anderson, Ross. *Security Engineering*. Second Edition, Wiley, 2006. Chapter 1: What is Security Engineering? pp. 3-15. Available at:

<http://www.cl.cam.ac.uk/~rja14/Papers/SEv2-c01.pdf>

**Lecture 5: Exercises**

1. As we have discussed, one of the benefits of representing information digitally is that errors introduced by random noise (or by a malicious attacks) can be detected and corrected. But how is this done? One simple and common mode for detecting errors in digital data is to add a *parity bit* to the data. This makes the data longer and adds *redundancy.* If noise causes a bit to be flipped, the redundancy can help reveal that error.

a. A typical way to calculate a parity bit is to set the extra bit so that the **total number ones in the word is even**. Add the parity bit to this 32-bit string:

1111 0101 1100 0111 0001 \_\_\_\_\_

b. Suppose you apply the XOR operation bitwise to a string of bits including the parity bit, what will be the result? (Applying XOR bitwise means XOR the first two bits, then XOR the result with the next bit, and so on).

c. Same scenario as above but now suppose one of the bits in the word is flipped by random noise, and you do the XOR operation to the altered string. Compare this result to the previous?

d. Suppose the parity bit indicates an error has occurred (that is the total number of ones in the word ends up as odd). Can you determine which bit was the source of the error?

e. What if two bits are flipped in a 32-bit word. Can a single parity bit detect the error?

f. What if parity were applied to each bit in the word (so each bit has a parity bit following it.) Can a single bit error be corrected?

g. Suppose now that you are conducting a “main in the middle” attack on a stream of messages, and you wanted to alter a particular byte from 0100 to 0000. Would you know how to reset the parity bit so that this change wouldn’t result in an error being detected?

2. Television images can be transmitted as a sequence of frames; each frame is like a single frame of a motion picture on film. But often the information in one frame is very similar to the previous one – only small parts of the image change.

Suppose now that if only the *differences* between successive frames are transmitted, rather than the whole frame. Is the amount of *data* transmitted (circle one)

a. greater b. the same c. less

than before?

Is the amount of *information* transmitted (circle one)

a. greater b. the same c. less

than before?

3. Give an example of a problem that is hard to solve, but whose solution is easy to check.

4. A cryptographic checksum attached to an unencrypted file of data that represents a software update received over an internet connection permits the recipient to determine (circle all that apply)

a. Whether the data has been observed by others

b. Whether the data has been modified after the checksum was signed

c. What *principal* (ref. Anderson, Security Engineering, p. 12) created the checksum of the file

d. What *subject* (ref. Anderson, Security Engineering, p. 12) sent the file of data

e. The update file contains no vulnerabilities

5. Computer structure:

a. Nearly all of today’s computers are based on the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

architecture, which incorporates

i. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ii. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

iii. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

b. My ability to place in my pocket today a computer with more power than one that occupied a full floor of equipment in 1970 was predicted by

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_