COMPSCI 314 S1T Assignment 2 2008

Department of Computer Science
The University of Auckland

*This assignment contributes **5%** of your overall course mark. Submit your assignment in **PDF** format to the **Assignment Drop Box**. Include all **workings** and **explanations**. Marks will be deducted for ambiguous solutions . Zero marks are awarded if the answers contain no explanation. Also, refer to the Departmental Policy on Cheating on Assignments.*

***Assignment Drop Box** (https://adb.ec.auckland.ac.nz/adb/).*
***Departmental Policy on Cheating on Assignments** (http://www.cs.auckland.ac.nz/CheatingPolicy.php)*

[Total: 50 marks]

Q1. Simple encryption [18 marks]

a) Write down your own University UPI. Then convert it to binary plaintext ***P*** using 8-bit ASCII code, as shown on Shay page 90 (plus an initial zero). If your UPI has only has 6 characters, add a space character at the end. [2 marks]

Examples:

abcd012 becomes
```
P = 01100001 01100010 01100011 01100100 00110000 00110001 00110010
```

abc012 becomes
```
P = 01100001 01100010 01100011 00110000 00110001 00110010 00100000
```

[Marker should just check a couple of the bytes - this question is intended to check understanding.]

b) Make up and write down a random 56 bit binary number. Call it ***k***.

Encrypt your binary UPI, using XOR (exclusive OR) as the encryption algorithm and ***k*** as the encryption key. Show your workings and write the encrypted binary cyphertext ***C***. [4 marks]

For example
```
  P = 01100001 0110010 01100011 01100100 00110000 00110001 00110010
  k = 01110010 1110100 01100111 11000110 10100100 10010101 00110011
  C = 00010011 1000110 00000100 10100010 10010100 10100100 00000001
```

[Marker should just check a couple of the bytes]

[Editing mistake - there is no c).]

d) What is the correct decryption algorithm and key? [3 marks]

Exactly the same as for encryption, i.e., symmetric key

e) Perform decryption. Show your result, and check that it is correct. [2 marks]

For example

```
 C = 00010011 1000110 00000100 10100010 10010100 10100100 00000001
 k = 01110010 1110100 01100111 11000110 10100100 10010101 00110011
 P = 01100001 0110010 01100011 01100100 00110000 00110001 00110010
```

Same as original

[Marker should just check a couple of the bytes]

f) How many keys would an attacker need to try, to be certain of breaking the cypher? [3 marks]

Approximately 2^{56} (or 72057594037927936) (or 1 less, since it isn't worth trying all 0s)

g) Would this be a good encryption method for users of an on-line shopping web site to authenticate the site (i.e., to be certain that they have connected to the genuine site)? Explain your answer. [4 marks]

No, because the key is symmetric (the same for encryption and decryption). So the key cannot be published, and a different secret key would be needed for each customer. A public/private key algorithm would be better.

Q2. Link efficiency. [18 marks]

Consider a data link from Auckland to Los Angeles, using an optical fibre connection on the seabed. Assume the distance is 10,000 km (10^7 m), and that the speed of an optical signal is $2x10^8$ m/s. Assume that the transmission capacity of the link is 1 Gb/s (10^9 bits/s), and that the frame size is 1500 bytes.

a) What is the transit delay for a data frame to travel from Auckland to Los Angeles or back? [2 marks]

$10^7/(2x10^8) = 50$ ms

b) For an approximate calculation, do we also need to consider the time taken for a computer to output a frame at 1 Gb/s? [3 marks]

No. 1500B at 1 Gb/s takes $(1500x8)/10^9$ s = 12 microseconds, okay to be ignored.

c) Consider a stop-and-wait protocol sending 1500 byte frames and waiting for an ACK after each frame. Assuming no frames or ACKs are lost, calculate the achieved bit rate in b/s and the efficiency of the protocol. [4 marks]

ACK will take 2x50 ms = 100 ms to return, hence we can send 1500B every 100ms, which is 1500x8x10 b/s = 120,000 b/s. Efficiency is 120000/10^9 = 0.0012%.
(It is okay to include both data+ack time (from b), the efficiency should still be similar.)

d) Calculate the bandwidth-delay product for the link, in megabytes. [3 marks]

1 Gb/s x 50 ms = 10^9 x 5 x 10^{-2} = 5 x 10^7 bits = 6.25 MB

e) For a sliding window protocol to use this link with high efficiency, what is a suitable window size (as a number of frames)? [3 marks]

2x6.25 MB / 1500 B = 8334 frames (enough time for the first ACK to come back).

f) How will your answer to e) change if the link is shared equally by 100 simultaneous users? [3 marks]

If 100 users share the link, they will each get a 1% share of the bandwidth, so they can use smaller window sizes, e.g. 83 frames.

Q3. Link layer [14 marks]

a) Why is the data link layer formally split into two layers, and what are they called? [4 marks]

Some data link functions like frame format, flow control and error checking don't depend on the hardware details. Others like addressing and contention mechanisms do depend on hardware. The two layers are Logical Link Control (LLC) and Medium Access Control (MAC).

b) Point-to-point protocols like HDLC assume that only two stations are sharing the medium (wire or wireless channel). Contention protocols assume many stations are sharing the medium.

Briefly describe two features that must be added to a link layer protocol to deal with three stations instead of two. [4 marks]

Some way to distinguish the stations (i.e., link layer addresses).
Some way to ensure that only one station transmits at a time (e.g., collision detection)

c) Is anything fundamentally different between three stations and any larger number of stations? [3 marks]

Perhaps collision rate may increase, but fundamentally, no.
If you can solve it for three, you've solved it for N.

d) In a contention protocol, when a collision between two frames is detected, is it OK if both stations wait the same length of time before retransmitting? Explain your answer. [3 marks]

No, because they will probably collide again.