THE UNIVERSITY OF AUCKLAND

FIRST SEMESTER, 2010
Campus: City

COMPUTER SCIENCE
Modern Data Communications
(Time allowed: TWO hours)

NOTE: Examination conditions: Closed book, no calculators permitted.

Attempt **ALL** questions. Total possible: **100 marks.**

This is an **ungraded sample** exam, which should take you about 15 minutes to complete.

Instead of using a script book, you should write your answers on a blank sheet of paper. Do **not** write your name on your answer sheet.

There will be a brief (approximately 5 or 10 minute) reading period at the beginning of the examination, during which you are **not** allowed to pick up a pen or pencil.

Sample answers from students are shown in blue.
Instructor's comments are shown in green.

A. The following questions refer to the series of four messages (M1-M4) described below.

M1. Alice → Bob: E_b("I'm Alice", a , t_a)

M2. Bob → Alice: E_a("Hi Alice, how can I help you?", t_a , t_b)

M3. Alice → Bob: E_b("Hi Bob, Please open a free email account for me.", a , t_a)

M4. Bob → Alice: E_a("No worries. Your username is a. Your password is x7ty23#", t_a , t_b)

1. If Eve is able to intercept messages, will she learn Alice's username (a) and initial password (x7ty23#) on Bob's email server? Explain briefly. **(3 marks)**

No, because she does not know Alice's private key so can not decrypt the message.

1.5 marks. I'm not sure which message the student is referring to. I'm also a little concerned that the student might believe that Eve must decrypt a message in order to learn Alice's public key. It seems quite possible that this student has not analysed the protocol, and has guessed that the expected answer is "no". Generally I don't give any credit for an unexplained answer, however in this case the student's answer would be correct if it had been a little more specific.

No, because the password is send by Bob to Alice, and not Alice to Bob. So, the Eve could read the Alice message but not then the Bob's send to Alice. She may find that is the username a without the password.

2 marks. This answer is quite specific – the student is clearly analysing the protocol. The student incorrectly asserts that an intercepting Eve can read messages that Alice encrypts under Bob's public key, correctly asserts that Eve can't read Bob's messages, and also correctly asserts that Alice's public key a is not a secret – so Eve may be able to learn it without reading messages M1 through M4.

Yes, because at last Bob had sent Alice both username and passwords. And also Eve will get the private and public key of Bob (ta, tb).

0.5 marks. The student seems to be assuming that all messages are sent in cleartext. The student has also assumed that Bob would reveal his private key to Alice – this would be very unwise, even if Bob trusts Alice.

Unless Eve knows Bob's private key, she will not be able to decrypt the initial message Alice sends even though she can intercept it. Hence she will not learn Alice's key, and will not be able to read or fool either party and cannot learn the username and password.

2.5 marks. This student's analysis is quite specific and plausible. I can't follow the logic in "Hence she will not learn Alice's key", because Alice's key is not revealed by this protocol; and I'm not sure what sort of attacks the student is thinking about in which Eve might "fool" Alice or Bob; however on the whole, the answer is quite good.

No. Because, even if Eve intercepts the message, she would need Alice's private key to decrypt the message that has been encrypted by Alice's public key by Bob (i.e. M4). Hence, unless Eve somehow miraculous hacks into Alice's computer or decodes the encryption, Eve will not be able to see the password or username.

3 marks. Specific and accurate.

2. If Eve is able to fabricate messages, will she be able to open an account on Bob's email server with username *a* which Alice doesn't know about? Explain briefly. **(4 marks)**

Eve could open an account if Bob has no checking for whether Alice's public key is actually Alice's i.e. Eb("I'm alice", e, te) the initial message sent by eve would be accepted if bob doesn't check.

4 marks. Excellent analysis, although it refers to a somewhat different attack that the one I was trying to describe in this question. In this student's attack, Eve manages to open an email account under her own public key. In the attack I had in mind, Eve is able to open an email account with username a (this is Alice's public key), but Eve is unable to read M4 (because does not know Alice's private key) so she won't learn the email password x7ty23#. Eve might be able to defraud (or at least to annoy) Alice with this new email account, for example by persuading other people that it is Alice's new email address – these people may then have difficulty contacting Alice.

Eve could possibly fabricate message to Bob if Bob did not require any sort of signature from Alice. Eve could send any public key to Bob and use that key to communicate with Bob pretending to be Alice. This is of course only possible if Bob's key is also known to Eve.

4 marks. Excellent.

B. Consider the following code:

e	0
t	1
i	00
a	01
n	10
m	11

4. Give three different possible decodings of 001. **(3 marks)**

eet, it, ea.

3 marks. Generally I don't give any credit for an answer without an explanation, however this is quite a trivial question so the explanation wouldn't be very informative.

5. If this code is extended by adding a parity bit to the end of each codeword, would the resulting code be prefix-free? Explain briefly. **(4 marks)**

Yes, only if the parity bit encodes for how many bits is encoding the first character sent. i.e. 0011 = ea, 0010 = it.

0 marks. The student seems to misunderstand the definition of a parity bit. They seem to think that a parity bit can indicate the length (1 or 2) of a codeword, as well as indicating the parity of that codeword. Their examples are not explained, but I gather that the student believes 001 is a codeword (even though all of the codewords in this table are of length 1 or 2).

No, I don't think so. Say you added 1. Then e and t can still be prefixes to longer codes. E.g. e for a and t for m. You need to make sure that none of the codes prefix the other codes.

3 marks. I am not confident that this student understands parity: it seems they think the parity bit will be 1 for "e"(codeword 0), i.e. odd parity; and that the parity bit will also be 1 for "t" (codeword 1), i.e. even parity. However I'm awarding good marks to this answer because I am very confident that this student understands the concept of prefix-free coding.

No. For example e 00, t 10, i 000. 6 of e could be mistake for two of i. Parity bits are useful to detect errors in fixed length codes.

3 marks. This student has constructed a nice example, which is easily understandable despite the arithmetic mistake: three "e"s, not six, could be confused with two "i"s. However, I am worried about this student's understanding of parity, for the parity bit in their example seems to be a constant (0) rather than either even or odd parity.

C. (Other questions). **[84 marks]**
