2910326 Computer security

Examiner's report: Zone A

General remarks

Many candidates performed well in this examination. In fact the examiners feel that although each question contained some unseen and applied material, the paper did not fully test the more able candidates. In next year's examination paper, candidates should therefore expect a larger quantity of unseen and harder question parts.

Some candidates lost marks through writing out entire sections from the subject guide. Candidates needed to apply theory and show a good understanding of the subject in order to gain high marks.

Following are comments on each of the examination questions and the two coursework assignments.

Question 1 (Password systems and access control)

This question was chosen by about half the candidates. Part a) was very well done. Although this was not a question for candidates who struggle with mathematics, for others it was an easy 10 marks. Candidates were expected to use their results from part i) to deduce that an exhaustive search would find a password based on the given policy on average in approximately 11 months. Therefore staff should have to change their passwords every 3 or 6 months (every month is too often and every year is too insecure).

Most candidates were able to interpret the given access control lists and represent them as an access control graph (see page 27 of the subject guide for an example of a similar graph). The graph is not a lattice since it contains several butterfly shapes. For example both S_1 and S_2 are connected to both O_2 and O_3. Therefore it is not clear whether S_1 or S_2 is dominant and there is no least upper bound.

Question 2 (DES and Key Escrow)

Few candidates who chose to answer this question scored high marks. For part a) each staff member will require *99* different keys in order to communicate with the 99 other members of staff. Therefore there are *(99*100)/2 = 4950* keys in total. Some candidates were confused about the difference between a symmetric key cryptosystem and a public key cryptosystem stating that each staff member would need two keys,one public and one private, to communicate with all the other staff.

Most candidates were able to give the blocksize and keysize for DES and 3-DES and explain how DES is transformed into 3-DES. However many candidates thought that 3-DES is three times safer than DES whereas it is many times more safe since the number of keys has increased exponentially. However 3-DES is three times slower than DES since three encryption/decryption steps are performed instead of only one.

For part c) the majority of candidates knew the purpose of a key escrow protocol but had difficulty in describing the *4 of 4* protocol for part i) and implementing the *2 of 4* protocol for part iii). Only the most able candidates set up the correct simultaneous equations ($5=5a + k \bmod 19$ and $13 = 9a + k \bmod 19$) and found the value of k. In this case subtracting the first equation from the second gives $8 = 4a \bmod 19$ and hence $a=2$. Substituting this value into the first equation gives $5=10+k \bmod 19$ and so $k=-5=14 \bmod 19$.

Question 3 (Algorithm for Exponentiation and RSA)

Candidates did very well on the bookwork parts of this question. However the application of this bookwork was not so well done. For part a) the majority of candidates could write the pseudocode for calculating x^n. Note that the question did not ask for modular exponentiation so '*mod n*' should not be included. However applying the algorithm to demonstrate how to find x^{43} was beyond many candidates. The algorithm works by squaring x and multiplying in other powers as necessary until x^n is reached. In this case we would calculate $x, x^2, x^3, x^4, x^8, x^{16}, x^{32}$ and x^{43}.

Similarly for part b) nearly all candidates were able to give details of the RSA pubic key cryptosystem and knew that they had to calculate $m=518^{13} \bmod 713$ for part ii). Applying the algorithm for exponentiation to calculate m did not prove so easy.

Question 4 (Diffie-Hellman, El-Gamal, hash functions)

Most candidates knew what is meant by saying that a function is **one-way** but few knew that the one-way function used as the basis for the security in the Diffie-Hellman key exchange protocol in the Discrete Logarithm Problem. Descriptions of the Diffie-Hellman protocol were variable. Most candidates had some idea of this protocol but few gave sufficient details. For example, it is important to begin by saying that Alice and Bob agree on a large prime p and generator g.

For part b) candidates were able to describe the key generation protocol for El Gamal or demonstrate it using the number provided. The best answers did both, for example starting off by saying 'First we generate a large prime, in this example $p=41$'.

Nearly all candidates could state the four essential properties of a cryptographic hash function but there was some confusion as to how to use a hash function and public key cryptosystem to sign a message. The important point is that the sender of the message uses his or her own private key to encrypt a hash of the message. This is the signature. The plaintext message is sent together with the signature. The receiver decrypts the signature using the sender's public key, hashes the message and compares these two results. If they match the message is authenticated.

Question 5 (PGP)

Part a) was very well answered. This part is pure bookwork and so is expected to be well done. It was pleasing to see that many candidates were also able to interpret the definitions and apply them sensibly to the given scenario of a security breach in part b).

Part c) was not so well answered. Although many candidates could give the PGP protocol for sending an encrypted, signed message, few were able to give adequate explanations of how PGP allows users to have more than one pair of keys.

Coursework 1 (Password Searching)

Students were asked to find six passwords given their encrypted values and the one-way function used for the encryption. They were asked to submit a report on how they tackled the problem of finding the passwords. A java program, which simulated the password input screen, was available for students to download.

Students tackled this problem in various ways. A sensible approach is as follows.

- First try password guessing; inputting very common passwords or words related to the coursework. The first password *cwk1* which was the name of the java program, may have been found using this method.
- Next try a dictionary search using a dictionary file downloaded from the Internet. This would reveal the second password *cat*.
- Modifications to the dictionary file such as changing to uppercase, mixing uppercase and lowercase, adding a digit at the end or changing a letter to a similar digit (1 for l for example) gained extra marks and may have found the passwords *Dogs, bus7* and *L0VE* (note that the 0 in L0VE is a zero not a letter O).
- Finally an exhaustive search should be attempted. This would reveal the final password *f9Z2*.

The report should have given clear explanations of students reasoning's for the methods that they used to find the passwords, timings of how long different searching methods took and an analysis/comparison of the different methods used. For example both a dictionary search and an exhaustive search would find the password **cats** but the dictionary search is likely to be much faster. That is why a real-life hacker would generally perform a dictionary search first. Some students stated that they only performed an exhaustive search because a dictionary search may only find one password whereas an exhaustive search would find them all. Although this is true, students should remember that generally a hacker need only find one password and then he or she can gain unauthorised access to the system. Students who only did an exhaustive search did not score highly for this assignment since although they may have found all of the passwords they could not compare different searching methods or write any meaningful analysis.

Coursework 2 (Application of R.S.A.)

This coursework was a practical application of the R.S.A public key cryptosystem. Most students were able to demonstrate use of the protocol by correctly encrypting and decrypting the given messages and ciphertexts. A problem arises when trying to use the given blocking method with both upper and lower case letters because some lower case letters have three rather than two digits in their ASCII representation. Most students commented on this, but many did not identity that the actual problem occurs when decoding the message. The receiver of the message will not know whether to take two or three digits of decrypted message to code into plaintext and so ambiguity arises as to what is the genuine message.

An alternative method of blocking suggested by many students is to pad all two digit ASCII codes with a zero so that all characters are represented by three digits. This solves the problem stated above. However if two characters are transmitted per block, the size of the block is now greater than the modulus n and therefore n needs to be increased. Many students

made the mistake of saying that they would transmit only one character per block. Although this method would definitely solve any ambiguity in the message, the effect it has is to transform RSA into a simple substitution cipher. Each letter is always encrypted to the same value. This obviously destroys the security of the cryptosystem.

The correct approach for part b) was to decrypt the ciphertext using your private key, decrypt the signature using Tom's public key and then compare the decrypted blocks. Any decrypted (message, signature) pairs that did not match were corrupted. In this way it was possible to deduce which blocks were not corrupt and hence find the message "MEET JIM AT 7". Since Charles has access to all public keys, he is able to decrypt the message if he intercepts the ciphertext simply by decrypting the signature blocks using Toms public key. Therefore this is not a secure method of transmitting a confidential signed message. There are several better methods.

Either Tom should first encrypt the message using the recipient's public key and then encrypt the ciphertext using his private key to generate the signature. He then sends both the ciphertext and the signature. Charles could decrypt the signature but he will only obtain a ciphertext, which requires a private key for decryption. The genuine recipient can decrypt the signature using Tom's public key, check that this matches the ciphertext to confirm that the message is from Tom, and then use their own private key to decrypt the ciphertext and regain the original message.

Alternatively Tom can use a hashing algorithm to produce a message digest of the original message. Then Tom encrypts this digest using his private key to generate a signature. He also encrypts the original message using the recipient's pubic key. He sends the ciphertext and signature. The genuine receiver can decrypt the ciphertext using their private key and hash the result using the hashing algorithm. Then they decrypt the signature using Tom's public key to regain the original message digest. If these two match then the message is authenticated.