Examiner's report 2009

326 Computer security – Zone A

General remarks

This exam paper proved to be very effective in distinguishing between those candidates who had simply memorised parts of the subject guide, and those who were able to apply what they had learnt to practical or unfamiliar settings, thereby demonstrating a real understanding of the subject. Computer security is really a practical subject so just being able to quote protocols and algorithms is not enough to gain a high mark in this examination. Although each question does contain some bookwork and is based on the material in the subject guide, to fully answer a question requires candidates to apply this material to a given situation. Many candidates were able to do this and demonstrated a good understanding of the subject. Note that common sense should also be applied and this was sometimes sadly lacking. Candidates should be careful to say what they mean and express themselves clearly.

Specific comments on questions

Question 1 (Security systems)

As some candidates stated, ideally a security system should provide all of the features listed in part (a)(i). However, this question is about recognising that in different situations, some features are more important than others. There is no definitive correct answer or correct order in which features should be listed. Marks were given to candidates who recognised the following points.

- Availability is very important. Subscribers have paid to access the site so they will soon become annoyed and perhaps withdraw their subscription if they cannot access it when they want to.
- Integrity is very important because the magazine does not want anyone to be able to make unauthorised changes to the site.
- Access control is important because the magazine only wants people who have paid the subscription fee to be able to access the site.
- Confidentiality is not important for the actual magazine site as the material is not confidential. However, subscribers are giving their credit card details to the magazine and these details should be kept confidential.
- Non repudiation is arguably the least important feature in this example. A subscriber could claim to have paid when in fact they haven't, or might have paid but the magazine says they haven't, but

this can be dealt with by good accountability. Recall that the subscription fee is only 'a small amount'.

- Accountability, as mentioned above, is important as the magazine needs to be sure who has paid to access the site and when their subscription expires. They also need to be able to see how much income they are making form the online site.
- Authentication is implemented by the username and password system. However, there is nothing to stop a subscriber from allowing someone else to access the magazine by giving them their username and password, so authentication is of limited use here.

Some candidates concentrated only on the credit card subscription part of the system and missed the point about the importance of the integrity and availability of the magazine itself.

There are many examples that could be used to answer part (a)(ii) but these should be **system features** such as:

- Log-on attempts should be limited – if a user inputs the incorrect password more than three times in a row they should be forced to reset their password as in the initial registration process.
- A subscriber can only log in at one computer at a time – this will prevent the subscriber from giving out their details to others who could then access the magazine without paying the subscription.
- Passwords which are less than seven characters long should be rejected – this will make exhaustive key space search harder.

Or interface features such as:

- Passwords should not appear on the screen but should be masked as ********* to prevent shoulder surfing.

But not **user features** such as 'the subscriber should choose a password which is an alphanumeric mix'. Note that suggestions for security features should always be proportionate to the item being secured. In this case there is no need for passwords of 15 characters or monthly password changes.

For part (b), the hacker uses each word in his dictionary to make ten possible passwords (e.g. apple0, apple1…apple9). Thus the hacker will try $10 * 10{,}000{,}000 = 10^8$ possible passwords. If he can try $1{,}000$ passwords per second then it will take him 10^5 seconds to try all of these passwords. However, given that the password is included in the key space that the hacker is testing, he will, on average only have to test half of the key space and so this will take $10^5/2 = 50{,}000$ seconds which is approximately 13.8 hours.

For part (c) again there is no definitive answer. Clearly if passwords are eight characters long then the key space is larger than if passwords are seven characters long. Therefore it can be argued that the first option is best. However in the second case, users are forced to choose passwords which contain at least two digits and so arguably these are more likely to be strong passwords and so the second option is best. There is a trade off here between length of password and strength of password. Full marks were given for discussing the pros and cons of each case.

Question 2 (Access control and protocols for secure transfer of information)

Part (a) of this question is bookwork. The graph given is not a lattice because Sales Manager 1 and Sales Manager 2 do not have a unique least upper bound. They are both dominated by Finance and Personnel but Finance does not dominate Personnel or vice versa. This creates a butterfly in the graph.

Adding the new security level Head of Sales in the position indicated for part (b) means that now Sales Manager 1 and Sales Manager 2 have a unique least upper bound (Head of Sales) and this also gives Finance and Personnel a unique greatest lower bound (Head of Sales). Therefore the graph is now a lattice and the security levels can be listed in hierarchical order as follows: Head > Finance = Personnel ≥ Head of Sales > Sales Manager 1 = Sales Manager 2> Shop 1=Shop 2 = Shop 3 = Shop 4 > Public.

Part (c) is about the differences between the secure transfer of electronic and physical material. To securely transfer the diamonds certain steps should be taken to ensure that the diamonds arrive safely, or that the shops are notified as soon as possible that something has gone wrong with the transfer. A possible solution is as follows. Shop 1 telephones Shop 4 to tell them to expect a package of diamonds and the approximate time of delivery. The diamonds should be locked into a secure case and the key to the case should not travel with the case between Shop 1 and Shop 4 (either the key is taken separately, or Shop 1 and Shop 4 have already got identical keys so that Shop 1 can lock the case and Shop 4 can unlock it). On arrival at Shop 4 (the diamonds have to travel physically either by secure courier or with a Shop 1 member) the identity of the Shop 4 staff must be verified (perhaps the Shop 1 member knows the Shop 4 member personally, or is able to identify them using a photograph) before the diamonds are handed over. Shop 4 telephones Shop 1 to say that the diamonds have arrived as expected.

Some candidates tried to apply the Diffie-Hellman key exchange protocol to answer this question which is not appropriate. Digital signatures do not apply either because the diamonds must be transferred physically. On the other hand, for part (ii) candidates were asked to design a protocol for the secure transfer of electronic information. The same principles apply as for part (i) – the protocol should ensure that the information arrives safely at its intended destination, or alert the Head and Finance offices if something has gone wrong. There are only insecure channels between Head and Finance so the information must be encrypted to ensure confidentiality. It must be digitally signed to ensure integrity. A protocol similar to the PGP protocol for confidentiality with authentication is suitable. As an additional precaution to detect modification of messages on the insecure channel, Finance and Head can talk to each other on the telephone to ensure that no keys have been tampered with during transmission. For example, Finance and Head agree on a cryptographically secure hash function and each generates their own set of public/private keys (for use with RSA for example). Head send their public key to Finance using the insecure channel. Then both Head and Finance hash this key. Over the phone they verify that they have

the same hash. If they have then Finance know that they have the correct public key for Head. The same thing happens to ensure that Head has the correct public key for Finance. If at any time either Head or Finance is expecting a message either by telephone or electronic transmission from the other office, and this message does not arrive, the office makes contact via telephone to find out the reason for the delay. If a message has been sent but has not arrived, then the teams assume that there is an attacker interrupting their communication and they abandon the protocol.

Question 3 (Encryption)

Part (a) is bookwork and most candidates were able to complete the sentences. In some cases there are several correct answers. The important thing is that the whole sentence is correct. For example both of the following sentences are correct but do not mix the two together. *It is important to have a large key space in order to prevent exhaustive search. It is important to have a large block size in order to prevent statistical analysis.*

There are two good reasons why it is desirable for a cryptosystem to use the same algorithm for encryption and decryption. The first is that encryption and decryption will take the same amount of time. This is important because otherwise, if for example decryption takes a longer time than encryption, it could cause a bottleneck in the process if lots of encryptions and decryptions are being performed. Differing lengths of time taken might also indicate to an eavesdropper whether it is an encryption or decryption taking place. The second reason is that if both algorithms are the same, developments in the efficiency of the algorithm will apply to both encryption and decryption and all efforts can be put into the development of a single algorithm rather than two different ones. This is obviously more cost effective. Single algorithm encryption schemes are also likely to be more suitable for use in 'small devices' with a limited amount of memory.

DES, the one-time pad, or substitution ciphers are all good examples of symmetric cryptosystems which have the same algorithms for encryption and decryption. RSA is an asymmetric cryptosystem which has this property.

For part (c) decryption is the inverse of encryption so instead of $C = M+K \bmod 27$ we have $M=C–K \bmod 27$. Further details are necessary for a full answer as follows. To decrypt a ciphertext, first associate each character in the ciphertext with the character at the corresponding position in the key (starting with the leftmost characters in each) repeat the key as many times as necessary to cover the entire ciphertext. Subtract the value of the key character from the value of the ciphertext character modulo 27. Convert the resulting number into a character as in table 1 (in the examination paper) to find the plaintext character. The given ciphertext decrypts to the message SHUT IT.

For part (c)(ii) marks were given for showing understanding of the following points.

If the message is shorter than the key then there is no repetition of the key. The encryption is similar to the one-time pad (assuming that the

same key is not reused for a different message). If the key is a random stream of characters rather than a word or phrase then this is equivalent to perfect secrecy in that an attacker can only guess the next part of the key with a one in 27 chance of success. Previous and subsequent parts of the key will not give away missing parts. However, unlike the one-time pad, the attacker may be able to intelligently guess the next part of the message using statistical attacks and the redundancy of the language.

If the message is longer than the key then the key must be repeated until the entire message has been encrypted. If the attacker knows (or can work out) the length of the key, he will be able to find matching pairs of key/ciphertext/plaintext and use this to his advantage. If the key is much shorter than the message and has to be repeated many times then the cryptosystem becomes very vulnerable to statistical analysis.

In both cases, the cryptosystem is much more secure if a long random key is used rather than a well-known word or phrase.

Question 4 (Public and private cryptosystems)

This was perhaps the most straightforward question but not one to attempt without a calculator. To answer the question fully requires knowledge of the key generation protocols for RSA and El Gamal, and how to encrypt, decrypt and digitally sign messages using public key cryptosystems. You must be able to apply the fast modular exponentiation algorithm and know that the time this takes is $O(b^3)$.

For part (a), first calculate $r = (p–1)(q–1)=42 * 22 = 924$. The private key $d = 425$ is given so it is required to find public key e such that $e * 425 = 1 \bmod 924$. Use trial and error (to work through the possible answers given) to find that $437 * 425 = 1 \bmod 924$ and hence $e = 437$.

To sign a message for part (b) calculate $s = m^d \bmod n = 12^{425} \bmod 989$ using the fast modular exponentiation algorithm. This will show that the signature $s = 292$.

The public El Gamal key $e = g^d \bmod p = 5^{13} \bmod 97$ for part (c). This is small enough to calculate directly on a calculator as $e = 1220703125 \bmod 97 = 29$.

To decrypt a ciphertext using El Gamal first compute $x = r^e \bmod p = 27^{13} \bmod 97 = 12$. Next use trial and error to find m where $c = (m * x) \bmod p$ so for part (d), $93 = (m * 12) \bmod 97$ so $m = 32$.

If n people are going to communicate using a symmetric key cryptosystem they will need a total of $n(n–1)/2$ keys, so for part (e) the 35 people will need $(35*34)/2 = 595$ keys.

Using a public key cryptosystem, each person will need only two keys (one public and one private key per person) and so now $2 * 35 = 70$ keys are required.

Modular exponentiation is an $O(b^3)$ algorithm. The numbers involved in part (g) are four times bigger and so it will take $4^3 = 64$ times as long to compute the modular exponentiation using numbers of size 300 bits. This is $64 *1.5 = 96$ seconds.

Question 5 (Diffie-Hellman)

To get a full ten marks for part (a) a good description of the Diffie-Hellman key exchange protocol is required. Details can be found in the subject guide. Remember to include a definition of the discrete logarithm problem and requirements made on the parameters in order to ensure that the discrete logarithm problem is a cryptographically secure one-way function, for example, p is a prime number of 100 or more digits, g is a generator for prime p and the secret keys a and b must be chosen so that $g^a > p$ (otherwise there is no discrete logarithm problem to solve). If the transmissions between Alice and Bob can be overheard but not modified or interrupted then the Diffie-Hellman key exchange protocol is secure because even if Charles knows x and y, he cannot find the values of a or b without solving a discrete logarithm problem. Nor can he find the value of k using the values of x and y – to do so is thought to be as difficult as solving a discrete logarithm problem.

If Charles can intercept the values of x and y sent between Alice and Bob and replace them both with the value 1 then the shared key $k = 1$ because $1^a = 1^b = 1 \bmod p$ whatever the values of a, b and p. To prevent this simple attack from being effective, Alice and Bob simply need to check that the values of x and y that they receive from each other are not equal to one. If either x or y is equal to one then Alice and Bob assume that they have an active attacker and abandon the protocol.

For part (c)(i) compute $z = 2^{10} \bmod 211$. This is small enough to compute on a calculator since $2^{10} = 1024 = 180 \bmod 211$.

For part (c)(ii) compute $K_{AC} = 5^{10} \bmod 211 = 123$, again this can be computed directly using a calculator.

For part (c)(iii) use the algorithm for fast modular exponentiation to compute $K_{BC} = 132^{10} \bmod 211 = 58$.

To make sure that their key exchange has not been manipulated by an attacker, Alice and Bob need to perform some kind of 'handshake' using the shared key K_{AB}. However, they cannot do this using the same channel as they used to perform the Diffie-Hellman protocol because then Charles could manipulate the handshake too. Instead, Alice and Bob could both hash the key using a pre-arranged secure hash algorithm. Then Alice telephones Bob and reads out the hash of the key. If this matches the hash of Bob's key then they are communicating with each other as expected. If the key hashes do not match then they have been attacked.

Some candidates said that Alice and Bob should digitally sign the values of x and y they send using RSA keys or similar. However, if Alice and Bob have access to RSA keys which they have confidence in, then why would they bother using the Diffie-Hellman key exchange protocol in the first place? Instead Alice could generate a key, encrypt it using Bob's public key, sign it using her own private key and then send it to Bob.