DUBLIN CITY UNIVERSITY

AUGUST/RESIT SOLUTIONS 2015/2016

MODULE: CA4005 - Cryptography and Security Protocols

PROGRAMME(S):

CASE - BSc in Computer Applications (Sft.Eng.)
ECSA - Study Abroad (Engineering and Computing)

YEAR OF STUDY: 4,X

EXAMINERS: Dr Geoff Hamilton (Ph:5017)
Dr. Ian Pitt

TIME ALLOWED: 3 hours

INSTRUCTIONS: Answer 5 questions. All questions carry equal marks.

PLEASE DO NOT TURN OVER THIS PAGE UNTIL INSTRUCTED TO DO SO

The use of programmable or text storing calculators is expressly forbidden.
Please note that where a candidate answers more than the required number of questions, the examiner will mark all questions attempted and then select the highest scoring ones.

Requirements for this paper (Please mark (X) as appropriate)

- □ *Log Tables*
- □ *Graph Paper*
- □ *Dictionaries*
- □ *Statistical Tables*
- □ *Thermodynamic Tables*
- □ *Actuarial Tables*
- □ *MCQ Only - Do not publish*
- □ *Attached Answer Sheet*

QUESTION 1 **[*Total marks: 20*]**

1(a) [5 Marks]

Block ciphers are usually designed to provide *confusion* and *diffusion*. Explain what is meant by each of these properties, and give examples of the features of block ciphers which are used to provide them.

Solution:
Confusion means that each bit of the ciphertext has a highly non-linear relationship with the plaintext bits and the key bits. Some features of block ciphers which are used to provide this are non-linear S-Boxes, the mixing of operations from different algebraic groups and data-dependent transformations.

Diffusion means that the effect of changing plaintext bits or key bits are spread and therefore affect many ciphertext bits. Some features of block ciphers which are used to provide this are P-Boxes, Feistel structures and pseudo-Hadamard transformations.

1(b) [10 Marks]

Describe the *Advanced Encryption Standard* (AES) with reference to the following (use diagrams if necessary):

- Encryption algorithm
- Decryption algorithm
- Block size
- Key size
- Number of rounds
- Robustness against attacks

Describe how AES provides confusion and diffusion.

Solution:
This is mostly bookwork, but some thought has to be put into inverting the encryption algorithm to implement decryption. Block size: 128. Key size: 128/192/256. Number of rounds: 10/12/14. AES is fairly robust against attacks.

AES provides confusion through its S-Box, which is generated by determining the multiplicative inverse in $GF(2^8) = \mathbb{Z}_2[x] \pmod{x^8 + x^4 + x^3 + x + 1}$, which is a non-linear function. It provides diffusion through the shift rows and mix columns operations.

1(c) [5 Marks]

What are the minimum recommended block size and key size which should be used for a block cipher? What attacks could be mounted against the block cipher if either of these sizes is less than recommended? What are the implications of this for AES?

Solution:
In the long-term, the block size and key size for a block cipher should be at least 128 bits. If the block size is less than recommended, then blocks will be repeated much more often, and attackers will be able to detect patterns in the output which leak secret information. If the key size is less than recommended, then the block cipher will be susceptible to a brute force attack over the smaller keyspace. AES uses

at least the minimum recommended key size and block size, so is secure against these attacks for the foreseeable future.

[End Question 1]

QUESTION 2 ***[Total marks: 20]***

2(a) [6 Marks]

A cryptographically secure hash function should be *pre-image resistant* and *collision-free*. Define these properties and why they are important for hash functions.

Solution:
A hash function is pre-image resistant if it is computationally infeasible to recover data from its digest. This is important because the original data should remain secure.

A hash function is weakly collision-free or second pre-image resistant if, given M, it is computationally infeasible to find a different M' such that $H(M) = H(M')$. It is strongly collision-free if it is computationally infeasible to find different messages M and M' such that $H(M) = H(M')$. This is important because being able to find collisions relatively easily allows an attacker to replace one message with another which they have found to have the same digest.

2(b) [7 Marks]

Describe how hash functions can be used to implement digital signatures. Explain why it is important in this context that a hash function has a digest size of at least 160 bits. Describe a simple fraud that could be perpetrated using digital signatures if the hash function digest size were only 64 bits.

Solution:
Rather than signing the original message, which may be computationally expensive, a hash function can be used to produce the message digest, which can then be signed. The receiver of the message with digest should apply the same hash function to the message to verify that the same digest is produced.

For a n-bit digest, on average $\sqrt{n}$ trials are required before a collision occurs, which we would like to avoid. For a 160-bit digest on average 2^{80} trials would therefore be required before finding a collision, which is computationally infeasible.

For a 64-bit digest, only 2^{32} trials would be required before finding a collision which is computationally feasible. A malicious participant A could therefore generate 2^{32} messages which are acceptable to another participant B and 2^{32} messages which are not acceptable, and find a collision between these two sets. If A sends the acceptable collision message to B, and B signs it, then A could later claim that B had actually signed the unacceptable collision message, since they will both have the same digest. Similarly, B could sign one message and later claim to have signed the other one.

2(c) [7 Marks]

Describe how hash functions can be used for message authentication. How do *Message Authentication Codes* (MACs) differ from *Manipulation Detection Codes* (MDCs)? Describe how a MAC can be constructed from a block cipher, and how a MAC can be constructed from a MDC.

Solution:
By sending the digest of a message along with the message itself, the integrity of the message can be checked by seeing whether applying the same hash function to the message gives the same digest.

A MDC is a hash function without a key, so the MDC of a message has to be sent over an authenticated channel to prevent tampering. A MAC is a hash function with a key, so the MAC of a message does not have to sent over an authenticated channel.

A MAC can be constructed from a block cipher by using CBC mode, with the output of the final block giving the MAC value. A MAC can be constructed from a MDC using a HMAC where $HMAC_k(m) = h(k||p_1||h(k||p_2||m))$, where k is the key, m is the message and p_1, p_2 are fixed strings used to pad k to a full block.

[End Question 2]

QUESTION 3 *[Total marks: 20]*

Consider a toy ElGamal example with prime modulus $p = 23$, generator $g = 4$ and private key $a = 10$.

3(a) [5 Marks]

Determine the public key value h.

Solution:
The public key value $h = g^a \pmod p = 4^{10} \pmod{23} = 6$.

3(b) [7 Marks]

Describe how encryption is done in ElGamal and encrypt the message $m = 19$ with the ephemeral key $k = 2$.

Solution:
Encryption in ElGamal is done by calculating $c_1 = g^k \pmod p$ and $c_2 = mh^k \pmod p$ and sending the ciphertext $c = (c_1||c_2)$.

For this example, $c_1 = 4^2 \pmod{23} = 16$ and $c_2 = 19 \times 6^2 \pmod{23} = 17$ so ciphertext $c = (16||17)$.

3(c) [8 Marks]

Describe how decryption is done in ElGamal and use this to decrypt the ciphertext $(18||19)$.

Solution:
Decryption in ElGamal is done by first using the private key a to compute $c_1^{p-1-a} \pmod p \equiv c_1^{-a} \equiv g^{-ak}$.

The message m can be recovered by computing $(c_1^{-a})c_2 \equiv m \pmod p$.

For this example, $c_1^{p-1-a} \pmod p = 18^{12} \pmod{23} = 18$.

So the message $m = (c_1^{-a})c_2 \pmod p = 18 \times 19 \pmod{23} = 20$.

[End Question 3]

QUESTION 4 ***[Total marks: 20]***

Consider the following protocol that allows entities A and B to mutually authenticate each other and to establish a shared session key.

1. $A \rightarrow B : A, N_A$
2. $B \rightarrow S : B, N_B, \{A, N_A\}_{K_{B,S}}$
3. $S \rightarrow A : N_B, \{B, N_A, K_{A,B}\}_{K_{A,S}}, \{A, B, N_B, K_{A,B}\}_{K_{B,S}}$
4. $A \rightarrow B : \{A, B, N_B, K_{A,B}\}_{K_{B,S}}, \{N_B\}_{K_{A,B}}$

4(a) [6 Marks]

Explain the role played by S and the use of the keys $K_{A,S}$, $K_{B,S}$ and $K_{A,B}$ in this protocol.

Solution:
S is a trusted third party that generates the shared session key and communicates this securely to both A and B. $K_{A,B}$ is the shared session key, $K_{A,S}$ is the long-term symmetric key that allows A to communicate securely with S, and $K_{B,S}$ is the long-term symmetric key that allows B to communicate securely with S.

4(b) [8 Marks]

Explain how the nonces N_A and N_B ensure the freshness of the session key, and their role in the mutual authentication of A and B.

Solution:
Nonces N_A and N_B are sent encrypted along with the session key, so cannot be replays from an earlier iteration of the protocol, and the session key must be fresh. A can authenticate B when she gets the reply from S in step 3 which is encrypted by her own secret key and contains her nonce N_A along with B's identity. B can authenticate A when he gets the message from A in step 4 which is encrypted using the new session key shared with A and contains his nonce N_B.

4(c) [6 Marks]

Consider the protocol obtained by replacing step 4 with the following step:

$4'$. $A \rightarrow B : \{A, B, N_B, K_{A,B}\}_{K_{B,S}}$

Does this new protocol still achieve mutual authentication? Explain your answer.

Solution:
This new protocol would not allow B to authenticate A. The message sent in this final step could simply be a replay of the second part of the message sent in the previous step.

[End Question 4]

QUESTION 5 ***[Total marks: 20]***

5(a) [7 Marks]

Describe the WEP protocol, including in particular how messages are encrypted and decrypted, and whether the values involved are public or private.

Solution:
WEP uses a long-term secret 128-bit key K and a public 24-bit initialization vector IV. It uses RC4 for confidentiality and CRC for integrity. RC4 is a stream cipher, so each packet must be encrypted using a different key. The actual RC4 key for a packet is $(IV||K)$ and the IV is sent with each packet.
The ciphertext c for message m is calculated as $c = \mathrm{RC4}(IV||K) \oplus (m,\mathrm{CRC}(m))$

5(b) [7 Marks]

Describe how message integrity is implemented in WEP. Explain using an example how an attacker can modify messages without detection. How could message integrity have been implemented to prevent this attack?

Solution:
A simple linear checksum has the property: $\mathrm{CRC}(x \oplus y) = \mathrm{CRC}(x) \oplus \mathrm{CRC}(y)$
Say the ciphertext $CT = \mathrm{RC4}(IV||K) \oplus (m,\mathrm{CRC}(m))$

The attacker creates $(m',\mathrm{CRC}(m'))$ and XORs this with the ciphertext to get:

$$\begin{aligned} CT' &= CT \oplus (m',\mathrm{CRC}(m')) \\ &= \mathrm{RC4}(IV||K) \oplus (m,\mathrm{CRC}(m)) \oplus (m',\mathrm{CRC}(m')) \\ &= \mathrm{RC4}(IV||K) \oplus (m \oplus m',\mathrm{CRC}(m \oplus m')) \end{aligned}$$

The attacker has changed the ciphertext in such a way that it will now decrypt to $m \oplus m'$, and the CRC will still be okay.

If the attacker wants to change the message to m'', then select m' s.t. $m \oplus m' = m''$ i.e. $m' = m \oplus m''$.

Message integrity could have been implemented using a cryptographic hash to prevent this attack.

5(c) [6 Marks]

WEP uses a 24-bit IV. If this IV were generated randomly for each packet, after how many packets would you expect the same IV to be generated? What information could this re-occurrence reveal to an attacker?

Solution:
By the birthday paradox, we would expect the same IV to be generated after on average 2^{12} packets.

If the long-term key K and the IV are the same, then the same keystream is used.

If $C_1 = \mathrm{RC4}(IV||K) \oplus P_1$ and $C_2 = \mathrm{RC4}(IV, K) \oplus P_2$ then $C_1 \oplus C_2 = P_1 \oplus P_2$

[End Question 5]

[END OF EXAM]