DUBLIN CITY UNIVERSITY

SEMESTER 1 SOLUTIONS 2017/2018

MODULE:	CA4005 - Cryptography and Security Protocols	
PROGRAMME(S):		
CASE	BSc in Computer Applications (Sft.Eng.)	
CPSSD	BSc in ComputationalProblem Solv&SW Dev.	
ECSAO	Study Abroad (Engineering and Computing)	
YEAR OF STUDY:	4,O	
EXAMINER(S):		
	Geoffrey Hamilton	(Ext:5017)
	Dr. Hitesh Tewari	External
	Prof. Brendan Tangney	External
TIME ALLOWED:	3 Hours	
INSTRUCTIONS:	Answer all questions.	

PLEASE DO NOT TURN OVER THIS PAGE UNTIL INSTRUCTED TO DO SO.
The use of programmable or text storing calculators is expressly forbidden.

There are no additional requirements for this paper.

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

1(a) [4 Marks]

Compare and contrast *stream ciphers* and *block ciphers*.

Solution:
Block ciphers encrypt one block of data at at time, while stream ciphers encrypt an arbitrary stream of data. Stream ciphers use simpler arithmetic, so tend to be more efficient, but block ciphers are more versatile and can also be used to implement stream ciphers.

1(b) [12 Marks]

Compare and contrast the block ciphers *Data Encryption Standard* (DES) and *Advanced Encryption Standard* (AES) with respect to the following (use diagrams if necessary):

- Encryption/decryption
- Block size
- Key size
- Number of rounds
- Robustness against attacks

Solution:
This is mostly bookwork, but some thought has to be put in to inverting the encryption algorithm to implement decryption. Block size: DES 64, AES 128. Key size: DES 56, AES 128/192/256. Number of rounds: DES 16, AES 10/12/14. DES is slightly vulnerable to linear and differential cryptanalysis attacks, and to brute force attacks; AES is much more robust against attacks.

1(c) [4 Marks]

Show how a block cipher can be used to implement a stream cipher.

Solution:
By operating a block cipher in OFB or CFB mode, a block cipher can be used to implement a stream cipher.

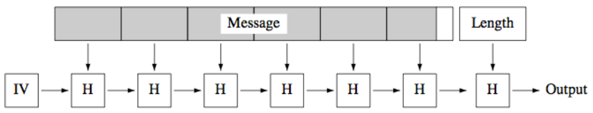
[End Question 1]

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

2(a) [8 Marks]

Describe the *Merkle Damgård construction* which is often used in the implementation of hash functions (use diagrams if necessary). What properties are required for a hash function to be considered to be cryptographically secure and why?

Solution:
The Merkle Damgård construction divides the message M into fixed-length blocks M_1, M_2, etc., pads the last block and appends the message length to the last block. We denote the resultant last block (after all paddings) by M_n. Then, the hash function applies a collision-free function H on each of the blocks sequentially. The compression function H takes as input the result of the application of H on the previous block (or a fixed initial value IV in the first block), and the block itself, and outputs a hash value. This hash value is an input to the application of H on the next block.



To be considered cryptographically secure, a hash function should be pre-image resistant and collision-free. 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 may need to be kept secret. 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 $H(M) = H(M')$. A hash function is strongly collision-free if it is computationally infeasible to find different messages M and M' such that $H(M) = H(M')$. These are 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) [8 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.

2(c) [4 Marks]

If a MAC with secret key k were created from a MDC as MAC$_k(m)$ = MDC$(k||m)$, show how you could make use of the Merkle Damgård construction to compute MAC$_k(m||m')$ without knowing k.

Solution:
The value of $\text{MAC}_k(m) = \text{MDC}(k||m)$ obtained previously can be fed back into successive applications of the compression function H to add the blocks for the additional message part m', thus giving the correct value for $\text{MAC}_k(m||m') = \text{MDC}(k||m||m')$

[End Question 2]

QUESTION 3 **[Total marks: 20]**

Consider a toy RSA example in which the public key is $(N = 55, e = 17)$.

3(a) [6 Marks]

Determine the value of the private key.

Solution:
The private exponent $d = e^{-1} \pmod{\phi(N)}$ i.e. $17^{-1} \pmod{40}$. This can be calculated using the extended Euclidean GCD algorithm:

$$\begin{aligned} 40 &= (17 \times 2) + 6 \\ 17 &= (6 \times 2) + 5 \\ 6 &= (5 \times 1) + 1 \end{aligned}$$

So:

$$\begin{aligned} 6 &= 40 - (17 \times 2) \\ 5 &= 17 - (6 \times 2) = 17 - (2 \times 40) + (4 \times 17) = (5 \times 17) - (2 \times 40) \\ 1 &= 6 - (5 \times 1) = 40 - (2 \times 17) - (5 \times 17) + (2 \times 40) = (3 \times 40) - (7 \times 17) \end{aligned}$$

So $17^{-1} \pmod{40} = -7 = 33 \pmod{40}$

The private key is therefore $(N = 55, d = 33)$.

3(b) [7 Marks]

Describe how a digital signature can be implemented using RSA. Describe a technique which can be used to perform the mathematical operation used in this digital signature more efficiently using the prime factors of the modulus, and use this technique to generate the digital signature for the message digest value 35.

Solution:
The digital signature can be calculated as $m^d \pmod{pq}$ and we can calculate this more efficiently using $m^d \pmod{p}$ and $m^d \pmod{q}$ and the Chinese Remainder Theorem.

To calculate $35^{33} \pmod{55}$, we calculate $35^{33} \pmod{5}$ and $35^{33} \pmod{11}$ and combine using the Chinese Remainder Theorem.

$35^{33} \pmod{5} = 0^1 \pmod{5} = 0$ and $35^{33} \pmod{11} = 2^3 \pmod{11} = 8$, so $35^{33} \pmod{55} = 30$

So the digital signature value is 30.

3(c) [7 Marks]

Describe how a digital signature can be verified using RSA. Give an efficient algorithm which can be used to perform both encryption and decryption in RSA, and use this algorithm in the verification of the digital signature value calculated above.

Solution:
Verification of digital signature value s is performed by calculating $m = s^e \pmod{N}$ and checking whether the result is the same as the received message digest value. An efficient algorithm for this modular exponentiation is the square and multiply algorithm; this can be computed bit by bit left-to-right or right-to-left. The left-to-right variant for computing $s^e \pmod{N}$ where e has n bits $e_{n-1} \dots e_0$ is as follows:

```
y = 1
for i = n-1 downto 0 do
   y = (y*y) mod N
   if $e_i$ = 1 then
      y = (y*s) mod N
   end
end
```

To verify the digital signature value 30, we need to compute $30^{17} \pmod{55}$. Using the described algorithm, this is computed as follows:

i	x_i	y
4	1	$1 \times 1 \times 30 \pmod{55} = 30$
3	0	$30 \times 30 \pmod{55} = 20$
2	0	$20 \times 20 \pmod{55} = 15$
1	0	$15 \times 15 \pmod{55} = 5$
0	1	$5 \times 5 \times 30 \pmod{55} = 35$

So the correct digest value 35 is obtained.

[End Question 3]

QUESTION 4 *[Total marks: 20]*

4(a) [10 Marks]

Describe the *Otway-Rees* protocol. Your answer should include the objectives of the protocol, the protocol steps, and an informal analysis of the security of the protocol.

Solution:
Objectives of the protocol:

1. To allow A and B to be authenticated.
2. To allow S to generate a session key $K_{A,B}$ that is bound to the authentication process and is only known to A, B and S.

Protocol Steps:

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

Analysis of the protocol:

- In this protocol both A and B perform mutual authentication with the server S and S returns the session key $K_{A,B}$ to each.
- In the case of A we have:
  - S can authenticate A by decrypting $\{N_A, N, A, B\}_{K_{A,S}}$ it receives in step 2.
  - A can authenticate S and obtain its copy of the session key by decrypting $\{N_A, K_{A,B}\}_{K_{A,S}}$ in step 4.
  - S cannot ensure that the message from step 1 is not a replay but this is okay, because an attacker replaying the message will be unable to recover the session key $K_{A,B}$ in step 4.
  - In this protocol, B relays messages to/from A. However, since these messages are encrypted using $K_{A,S}$, B cannot decrypt or modify them.
- In the case of B we have a similar argument with $\{N_B, N, A, B\}_{K_{B,S}}$ and $\{N_B, K_{A,B}\}_{K_{B,S}}$

4(b) [5 Marks]

Explain the use of nonces in the Otway-Rees protocol. Does the Otway-Rees protocol require synchronized clocks? Explain your answer.

Solution:
In step 1, A generates two nonces N_A and N:

- N_A which is used to prevent replay of S's message back to A and guarantee the freshness of the session key received from S.
  - The nonce N_B serves the same purpose for B.
- N which acts as a transaction identifier that binds the information sent by A and B together:
  - S can check that $\{N_A, N, A, B\}_{K_{A,S}}$ and $\{N_B, N, A, B\}_{K_{B,S}}$ have the same value for N and therefore belong to the same transaction.

The Otway-Rees Protocol does not require synchronized clocks because it uses nonces rather than timestamps to provide freshness guarantees; to determine whether a timestamp is fresh would require the use of synchronized clocks.

4(c) [5 Marks]

What is a *type flaw attack*? Explain why the Otway-Rees protocol may be vulnerable to a type flaw attack.

Solution:
A type flaw attack on a security protocol is an attack where a field that was originally intended to have one type is subsequently interpreted as having another type.

A potential type flaw attack on the Otway-Rees protocol is as follows:

1. $A \to E(B): \quad N, A, B, \{N_A, N, A, B\}_{K_{A,S}}$

4. $E(B) \to A: \quad N, \{N_A, N, A, B\}_{K_{A,S}}$

This is valid if A can be fooled into believing that the triple N, A, B is a key.

[End Question 4]

QUESTION 5 [Total marks: 20]

5(a) [7 Marks]

Define what is meant by a *blind signature* and describe how this can be implemented using RSA. Explain how a blind signature scheme can be used to implement digital cash (this should include a description of the steps involved in a digital cash transaction).

Solution:
A blind signature is a mechanism to allow a message to be signed without revealing the contents of the message. Using RSA, with modulus n, encryption exponent e and decryption exponent d, A can obtain B's signature on message m as follows:

- A generates a random value r such that $\gcd(r, n) = 1$ and sends $x = (r^e \cdot m) \pmod n$ to B.
- The value x is *blinded* by the random value r; hence B can derive no useful information from it.
- B returns the signed value $t = x^d \pmod n$ to A.
- Since $x^d \equiv (r^e \cdot m)^d \equiv r \cdot m^d \pmod n$, A can obtain the true signature s of m by computing $s = r^{-1} \cdot t \pmod n$

To implement digital cash, the bank can apply blind signatures to messages which correspond to a monetary value.

The steps involved in a withdrawal by A are as follows:

1. A chooses a_i, c_i, d_i, $r_i \in Z_n^*$, for $i = 1, \dots, k$
2. A forms a coin: $C_i = f(g(a_i, c_i), g(a_i \oplus (u||(v + i)), d_i))$ (f and g are collision-resistant hash functions, u is A's account number and v is a counter)
3. A sends $r_i^e \times C_i$ to the bank
4. The bank picks a set of $k/2$ indices, R, and sends them to A
5. A sends a_i, c_i, d_i, and r_i for $i \in R$ to the bank
6. The bank produces a signature on the remaining C_i's: $s' = \prod_{i \notin R} r_i \times C_i^d$
7. A generates the final coin: $C = s' / \prod_{i \notin R} r_i = \prod_{i \notin R} C_i^d$
8. The bank increments v by 1, debits A's account by €1

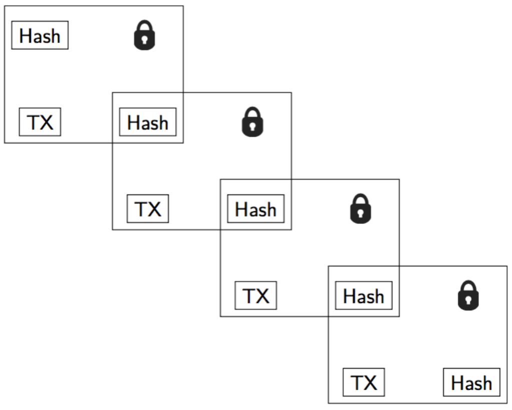
The steps involved in a payment by A to B are as follows:

1. A sends C to B
2. B chooses $k/2$ random bits, $z_1, \dots, z_{k/2} \in \{0, 1\}$
3. For each i, A sends:
   (a) If $z_i = 1$, she sends a_i, c_i, $g(a_i \oplus (u||(v + i)), d_i)$
   (b) If $z_i = 0$, she sends $g(a_i, c_i)$, $a_i \oplus (u||(v + i))$, d_i
4. B recomputes each C_i and verifies that the signature is correct
5. Later, B sends C and A's responses to the bank
6. Bank verifies the responses and credits B's account

5(b) [7 Marks]

Define the structure of the *blockchain* used in Bitcoin, and explain how this is used to implement digital cash (this should include a description of the steps involved in a digital cash transaction).

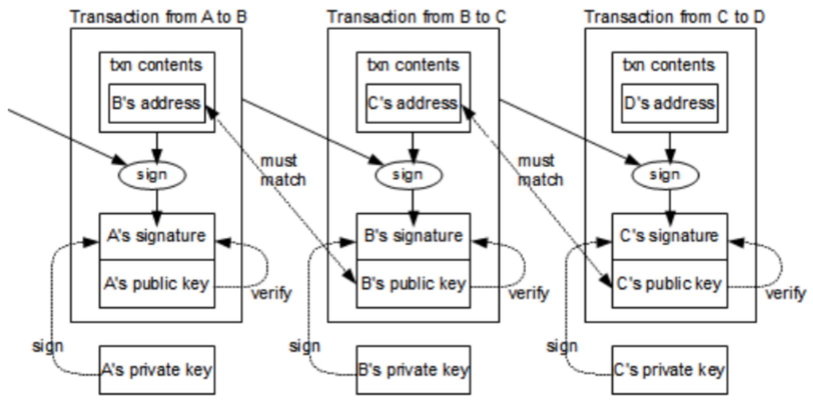
Solution:
The blockchain consists of a linked chain of blocks, each of which contains a number of transactions. Each block contains the hash of the previous block, thus providing a backward link in the chain. Each block gives security to the previous ones:



The steps involved in a payment by A to B are as follows:

1. A digitally signs the details of the transaction.
2. B broadcasts this transaction.
3. Miners validate the transaction within a block, along with a proof of work.
4. When the transaction has been validated 6 times it is fully confirmed.

Transactions have the following structure:



5(c) [6 Marks]

What are the main security objectives required of a digital cash scheme? Explain how these objectives are met by the digital cash schemes described in your answers for parts (a) and (b) of this question.

Solution:
The main security objectives of a digital cash scheme are as follows:

1. Secure transfer in computer networks
2. No double spending
3. Anonymity
4. Can be transferred to others
5. No forgery

For blind digital signatures, these objectives are met as follows:

1. Secure transfer in computer networks: all transactions are digitally signed.
2. No double spending: this is ensured using the cut-and-choose method.
3. Anonymity: blind signature makes signed coin unlinkable to owner, and identity is hidden in coin by hashing.
4. Can be transferred to others: coin can be redeemed by the recipient by sending the coin and the challenge responses to the bank.
5. No forgery: cannot fake the digital signature of the bank.

For Bitcoin, these objectives are met as follows:

1. Secure transfer in computer networks: all transactions are digitally signed.
2. No double spending: this is ensured by the method of validation fo transactions.
3. Anonymity: the only identity visible is the wallet identifier, and wallets can be replaced.
4. Can be transferred to others: transactions involve the transfer of Bitcoin.
5. No forgery: counterfeit Bitcoin will not be validated.

[End Question 5]

[END OF EXAM]