TTM4135 May exam 2016:
Outline answers

Exercise 1 Multiple choice questions

1.	(a) ☐	(b) ☐	(c) ☐	(d) ☑
2.	(a) ☐	(b) ☑	(c) ☐	(d) ☐
3.	(a) ☐	(b) ☐	(c) ☐	(d) ☑
4.	(a) ☑	(b) ☐	(c) ☐	(d) ☐
5.	(a) ☐	(b) ☑	(c) ☐	(d) ☐
6.	(a) ☐	(b) ☐	(c) ☐	(d) ☑
7.	(a) ☐	(b) ☐	(c) ☑	(d) ☐
8.	(a) ☐	(b) ☐	(c) ☑	(d) ☐
9.	(a) ☐	(b) ☐	(c) ☑	(d) ☐
10.	(a) ☐	(b) ☐	(c) ☑	(d) ☐
11.	(a) ☐	(b) ☐	(c) ☑	(d) ☐
12.	(a) ☐	(b) ☑	(c) ☐	(d) ☐
13.	(a) ☐	(b) ☐	(c) ☐	(d) ☑
14.	(a) ☐	(b) ☐	(c) ☐	(d) ☑
15.	(a) ☑	(b) ☐	(c) ☐	(d) ☐
16.	(a) ☐	(b) ☐	(c) ☐	(d) ☑
17.	(a) ☐	(b) ☐	(c) ☐	(d) ☑ [1]
18.	(a) ☐	(b) ☐	(c) ☑	(d) ☐
19.	(a) ☑	(b) ☐	(c) ☐	(d) ☐
20.	(a) ☐	(b) ☐	(c) ☑	(d) ☐
21.	(a) ☑	(b) ☐	(c) ☐	(d) ☐
22.	(a) ☑	(b) ☐	(c) ☐	(d) ☐
23.	(a) ☐	(b) ☑	(c) ☐	(d) ☐
24.	(a) ☑	(b) ☐	(c) ☐	(d) ☐
25.	(a) ☐	(b) ☑	(c) ☐	(d) ☐

[1] In the grading for Q.17 all answers except (c) were awarded 1 mark. While (d) is clearly the best answer, other answers are arguably correct too.

Exercise 2 Written answer questions

1. (a) There are 2^{64} keys in total so, using the approximation that there are 2^{25} seconds in a year, the attacker must search through $2^{64}/2^{25} = 2^{39}$ keys per second.
   (b) The correct key can typically be found in practice using plaintext redundancy as long as the plaintext is in a known and redundant form, such as natural language. Another likely method is a known plaintext attack where the attacker has a plaintext-ciphertext pair and can wait for the known plaintext to appear while checking all possible decryption keys on the ciphertext.
   (c) There are $2^8 = 256$ times as many keys now. Thus at the same rate it will take the attacker 256 years. However, due to improvements in hardware (as predicted by Moore's Law) this will not be realistic in practice. After only 20 years computers can be expected to be 1000 times faster.

2. (a) From the encryption equation, $C_2 = E(P_2 \oplus C_1, K)$ we see that changing P_2 will result in a random change to C_2 as long as the encryption algorithm is a strong block cipher. Moreover this will then propagate to $C_3 = E(P_3 \oplus C_2, K)$ which will also be randomly changed. Similar for C_4. Thus the new ciphertext can be expected to be competely different for C_2', C_3', C_4' while $C_1' = C_1$ since its inputs are unchanged.
   (b) The decryption equation is $D(C_t, K) \oplus C_{t-1} = P_t$. Thus a bit flip in C_2 can be expected to randomly change P_2' but only flips one bit in P_3'. However, P_1' and P_4' are unchanged.

3. (a) Since 17-1 has only 2 as the only prime factor, we only need check that $g^8 \bmod 17 \neq 1$ in order to verify that g is a generator.
   $2^8 \equiv 4^4 \equiv 16^2 \equiv -1^2 \equiv 1 \bmod 17$.
   $3^8 \equiv 9^4 \equiv 81^2 \equiv -4^2 \equiv 16 \equiv -1 \bmod 17$.
   (b) A computes $y^a \bmod p = 8^3 \bmod 17 = 2^9 \bmod 17 = 2$.

4. An RSA signature on a message m is a pair (m, s) where $s = h(m)^d \bmod n$, h is a suitable hash function, and n is the modulus which is part of the public key (e, n).
   (a) The recipient computes $s^e \bmod n$ and checks that it equals $h(m)$.
   (b) h must be collision resistant otherwise the attacker can obtain a signature on one message which will be a valid signature for a different message with the same hash value.
   (c) Choose a random signature value s and compute $H = s^e \bmod n$. Then (s, m) is a valid signature where $m = H - 1 \bmod n$ since $h(m) = H$.

5. (a) MAC and digital signature both provide message integrity and authentication. They both can be computed on a message of arbitrary size (in practice).
   MAC is a symmetric key algorithm while signature is asymmetric. Signatures provide the non-repudiation property since only the private key owner can sign. MAC does not provide non-repudiation since any party with the shared key can generate the MAC tag.
   (b) TLS uses MAC (usually HMAC) to protect data integrity in the record layer (user payload data) and can use signatures in the handshake protocol to sign the exchanged Diffie-Hellman value. Signatures are also used to validate public key certificates.
   Signatures could in principle be used in the record layer but would be inefficient. MAC could not be used in the handshake protocol unless client and server have a pre-shared key.

6. Since B does not receive any freshness guarantee from S the replay must be against him. The attacker can eavesdrop an old token $T' = \{K'_{AB}, A, B, N_A\}_{K_{BS}}$, the value N_A, and obtain the old key K'_{AB}. Then the attacker C masquerades as both S and A and starts a new session with the following messages.

   3. $S(C) \rightarrow B \quad : \quad T'$
   4. $B \rightarrow A(C) \quad : \quad \{ID_B, ID_A, N_A, N_B\}_{K'_{AB}}$
   5. $A(C) \rightarrow B \quad : \quad \{ID_A, ID_B, N_A, N_B\}_{K'_{AB}}$

   C takes the roles of both A and S and can compute the correct message 5 since it has the session key and can therefore decrypt message 4 to obtain N_B. The exchange looks normal to B who accepts the old key K' .

7. (a) The attack would be that the adversary obtains the long-term keys of both parties and uses that to recover previously agreed session keys.

   (b) The web server must only allow ciphersuites which use ephemeral Diffie-Hellman for the handshake protocol.

   (c) Since there is no interaction it must be possible for the mail recipient to decrypt the message with the long-term key alone. Therefore forward secrecy (strictly speaking) can never be achieved for email.