TTM4135 exam May 2021: Outline answers

1 Multiple choice questions

1 point for correct answer, 0.5 point penalty for incorrect answer. Explanation can say why one answer is correct, **or** why two answers are wrong for 1 point. Possible to get 0.5 point for explanation of why one answer is wrong.

1. Suppose that $x^{-1} \bmod 17 = 5$. Then

   (a) $x \bmod 17 = 7$ ✓
   (b) $x \bmod 17 = 6$
   (c) $x \bmod 17 = 5$

   Explanation: $5 \times 7 \bmod 17 = 35 \bmod 17 = 1$

2. Suppose that the 26-letter alphabet, $A \ldots Z$ is used for the plaintext in the 2×2 Hill cipher. Suppose that the letter E is the most common letter in the plaintext, occuring with frequency equal to 10%. Then in the ciphertext we can expect that:

   (a) the most common letter occurs with frequency equal to 10%
   (b) the most common letter occurs with frequency below 10% ✓
   (c) the most common letter occurs with frequency above 10%

   Explanation: Since the Hill cipher is simple substitution in pairs, each occurrence of the common characters can be encrypted in many ways, thereby smoothing the frequencies.

3. A typical RSA private key in use today may have length 3072 bits, but a typical symmetric key for the AES block cipher may have length only 128 bits. This longer key for RSA is necessary because:

   (a) security for public key encryption needs to be stronger than for symmetric key encryption
   (b) RSA keys need to be longer than symmetric keys to avoid attack by quantum computers
   (c) there are much better ways to attack RSA than brute force key search ✓

   Explanation: Factorisation has sub-exponential time algorithms, so that factorising the modulus and then computing the private key directly from the public key, is much easier than brute for search for the private key.

4. Suppose that in a binary synchronous stream cipher a section of the ciphertext is 01000. An attacker knows that the plaintext used to obtain this ciphertext is 00101. The corresponding section of the decryption keystream is:

   (a) 01101 ✓
   (b) 00101
   (c) 01000

Explanation: Since $C = P \oplus K$ we can also calculate $K = P \oplus C = 00101 \oplus 01000 = 01101$.

5. Consider the version of the triple DES (3-DES) block cipher with three independent keys. Compared with the AES block cipher, this version of 3-DES:

   (a) has fewer possible keys than all versions of AES

   (b) has a shorter block length than all versions of AES ✓

   (c) is faster to run in software than all versions of AES

   Explanation: 3-DES has a 168-bit key length, so not shorter than AES 128, but has a 64-bit block length, so shorter than AES.

6. Suppose that you have a message of 100 bits to encrypt and you choose to use the AES block cipher. Which of the following modes of operation will require the least number of sent bits in the encrypted message?

   (a) ECB mode ✓

   (b) Counter mode with a nonce of 64 bits

   (c) CBC mode

   Explanation: ECB uses one 128-bit block; counter mode uses 100 bits + 64 for nonce; CBC uses 128 for ciphertext block and 128 bits for IV.

7. The Euler function ϕ is often useful for public key cryptography. It is true that:

   (a) $\phi(n)$ is always divisible by 3

   (b) if n is divisible by 3 then $\phi(n)$ is always divisible by 3

   (c) if n is divisible by 9 then $\phi(n)$ is always divisible by 3 ✓

   Explanation: $\phi(3) = 2$ rules out the first two options. If n is divisible by 3^2 then $2 * 3$ is a factor of $\phi(n)$.

8. Suppose you want to prevent an attacker from finding a collision in a hash function. The attacker has enough computing power to calculate 2^{40} hash values. You need to ensure that the attacker has only small chance of success but prefer the smallest acceptable output size. You have three possible output sizes to choose from. Which should you choose?

   (a) 40 bits

   (b) 64 bits

   (c) 128 bits ✓

Explanation: Due to the birthday paradox, with 2^{40} trials there is a good chance of a collision being found with an output of length even 80 bits. Therefore more than 80 bits is necessary

9. A message authentication code, MAC, takes as input a key K and message M and outputs a tag T. In order to be secure, it is essential that:

   (a) an attacker who knows a valid M and T cannot find K ✓

   (b) an attacker who knows a valid K and T cannot find M

   (c) an attacker who knows a valid K and M cannot find T

   Explanation: Knowing K allows forgeries. Therefore no secure MAC can leak K from knowledge of T and M which will be known to an attacker.

10. The RSA signature scheme uses a modulus n and a public exponent e. If the modulus is chosen to be $n = 13 \times 23 = 299$ (**corrected**) then the smallest valid choice for e would be

    (a) $e = 3$

    (b) $e = 5$ ✓

    (c) $e = 7$

    Explanation: $\phi(243) = 2 \times 12 \times 11$, so is divisible by 2, 3 and 4. The value of e must be prime to $\phi(n)$ for the private key to exist.

11. For efficiency reasons it is often useful to keep fixed parameter values for many users of a cryptographic scheme. Which of the following is *not* a practical choice for digital signatures?

    (a) RSA signatures with a fixed modulus n ✓

    (b) DSA signatures with fixed generator g and fixed modulus p

    (c) ECDSA signatures with a fixed elliptic curve group

    Explanation: Knowledge of one RSA signing/verification key is enough to factorise n. It is normal to re-use the parameters for DSA and ECDSA.

12. When assessing the security of a key establishment protocol, such as the Needham–Schroeder protocol, we assume that an attacker is able to:

    (a) re-send messages sent in any previous runs of the protocol ✓

    (b) force parties to re-use nonces used in previous runs of the protocol

    (c) obtain long-term keys used in any previous runs of the protocol

Explanation: We assume that the network is controlled by the attacker who can then send any known values to the users. Nonces are chosen by users and cannot be controlled by the attacker. Long-term keys usually are static and a protocol cannot be secure in the future once long-term keys are known.

13. In the TLS 1.2 handshake protocol, a ciphersuite is negotiated between the client and the server. Which of the following does *not* depend on the chosen ciphersuite:

    (a) the algorithm used to authenticate the record layer data

    (b) the algorithm used to sign the server key exchange message

    (c) the algorithm used to sign the server certificate ✓

    Explanation: Only the algorithms used actively in the handshake and record protocol are negotiated. The certificate is signed beforehand.

14. TLS 1.3 aims to establish secure connections faster than TLS 1.2. One difference between the protocols which contributes to this is:

    (a) clients can send a Diffie–Hellman ephemeral value before the ciphersuite is agreed ✓

    (b) checking of server certificates is no longer required

    (c) servers can initiate the handshake protocol and use a ciphersuite of their choice

    Explanation: TLS 1.3 allows optimistic sending of the client key exchange message in the first flow, before the ciphersuite is decided.

15. PGP is a security protocol to protect emails in transit. PGP has seen very limited usage in practice. One of the reasons for this is:

    (a) usability is a challenge for many potential users ✓

    (b) encryption is provided but it is not possible to authenticate mail senders

    (c) PGP-encrypted mail cannot be sent on the normal email system

    Explanation: PGP is transparent to mail servers and has optional signatures. It is notoriously hard for the average user to configure.

2 Written answer questions

1. Suppose that you share a new (unused) random key of 128-bits with a recipient. You are considering whether to use the key either as a one-time pad or with the AES block cipher in ECB mode.

   (a) Suppose first that you have a single message to encrypt, written in English as 16×8-bit bytes to make 128 bits in total. For this part assume that the key is used only once for this message. Compare the security of each of the two choices. Is one better than the other and why?

   (b) Now suppose that you have a second message to encrypt, also written in English as 16×8-bit bytes. You decide to use the same encryption method with the same key as you used for the first 128-bit message. Again, compare the security of the two choices.

   One time pad gives perfect secrecy. For a single block, ECB mode is also secure except against brute force key search. When used twice, the OTP is no longer secure, and in fact by XORing the two blocks the XOR of the two messages is obtained. Using ECB is also leaking some information, in particular if they are the same or different.

2. The Feistel construction for a block cipher uses the round equations:

$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_{i-1} \oplus f(R_{i-1}, K_i) \end{aligned}$$

   for some function f. Suppose that f is chosen to be the function:

$$f(R, K) = R \oplus K$$

   for any half-block, R, and any round key, K.

   (a) Show that with this choice of f it follows that for all $i \geq 2$, both of the following equations hold.

$$\begin{aligned} R_i &= L_{i-2} \oplus K_{i-1} \oplus K_i \\ L_i &= L_{i-2} \oplus R_{i-2} \oplus K_{i-1} \end{aligned}$$

   We have from the Feistel equations:

$$\begin{aligned} R_i &= L_{i-1} \oplus R_{i-1} \oplus K_i \\ &= R_{i-2} \oplus L_{i-2} \oplus R_{i-2} \oplus K_{i-1} \oplus K_i \\ &= L_{i-2} \oplus K_{i-1} \oplus K_i \end{aligned}$$

$$\begin{aligned} L_i &= R_{i-1} \\ &= L_{i-2} \oplus R_{i-2} \oplus K_{i-1} \end{aligned}$$

(b) Use the above observation to show how to break a 2-round Feistel cipher with this f function given one known plaintext/ciphertext pair.

Given $C = (L_2, R_2)$ we know that $L_2 = L_0 \oplus R_0 \oplus K_1$ and $R_2 = L_0 \oplus K_1 \oplus K_2$. Since $P = (L_0, R_0)$ is also known, this allows the round keys K_1 and K_2 also to be found, so that any other ciphertexts can be decrypted.

(c) Explain, just giving the idea, how part (ii) can be generalised to break a Feistel cipher with any even number of rounds if this f function is used.

Each (L_i, R_i) pair can be replaced with (L_{i-2}, R_{i-2}) plus a fixed round key combination. This can be done recursively to get back to $P = (L_0, R_0)$.

3. One non-trivial square root of 1 modulo 209 is 153.

Alternate question:

One non-trivial square root of 1 modulo 221 is 118.

(a) What are all four of the square roots of 1 modulo 209?

The trivial square roots are 1 and -1. Others are 153 and $-153 \bmod 209 = 56$.

Alternate question:

What are all four of the square roots of 1 modulo 221?

The trivial square roots are 1 and -1. Others are 118 and $-118 \bmod 221 = 103$.

(b) Choose one of your non-trivial square roots, x and show, using the Euclidean algorithm, that $\gcd(x + 1, 209) > 1$.

Find $\gcd(57, 209)$.

$$\begin{aligned} 209 &= 3 \times 57 + 38 \\ 57 &= 38 + 19 \\ 38 &= 2 \times 19 \end{aligned}$$

So $\gcd(57, 209) = 19$ which is a divisor of 209

Alternate question:

Choose one of your non-trivial square roots, x and show, using the Euclidean algorirhm, that $\gcd(x + 1, 221) > 1$.

Find $\gcd(104, 221)$.

$$\begin{aligned} 221 &= 2 \times 104 + 13 \\ 104 &= 8 \times 13 \end{aligned}$$

So $\gcd(104, 221) = 13$ which is a divisor of 221

(c) Explain how an efficient algorithm to find non-trivial square roots can be used to break the RSA cryptosystem.

Given the RSA modulus n, the above process can be used to factorise n given a non-trivial square root. This allows $\phi(n)$ to be computed and the private key to be found from the public key.

4. The normal RSA cryptosystem uses modulus $n = pq$, a decryption exponent, d, and public exponent, e. Suppose that a company wants to protect its private exponent so that no single entity can decrypt. The manager splits d into two parts, d_1, d_2 such that

$$d_1 + d_2 \bmod \phi(n) = d.$$

In order to decrypt a ciphertext C, entity E_1 computes $M_1 = C^{d_1} \bmod n$, entity E_2 computes $M_2 = C^{d_2} \bmod n$ and these are combined to form $M = M_1 \times M_2 \bmod n$.

Alternate question:

The normal RSA cryptosystem uses modulus $n = pq$, a decryption exponent, d, and public exponent, e. Suppose that a company wants to protect its private exponent so that no single entity can decrypt. The manager splits d into two parts, d_1, d_2 such that

$$d_1 \times d_2 \bmod \phi(n) = d.$$

In order to decrypt a ciphertext C, entity E_1 computes $M_1 = C^{d_1} \bmod n$, entity E_2 computes $M = M_1^{d_2} \bmod n$.

(a) Show that a ciphertext encrypted with normal RSA, with public key e and n, is decrypted properly with this method. (You may assume that normal RSA works correctly.)

$M_1 \times M_2 \bmod n = C^{d_1} \times C^{d_2} \bmod n = C^{d_1+d_2} \bmod n = C^d \bmod n = M$, applying Euler.

$M_1^{d_2} \bmod n = (C^{d_1})^{d_2} \bmod n = C^{d_1 d_2} \bmod n = C^d \bmod n = M$, applying Euler.

(b) This system runs slower than normal RSA. To improve the efficiency the manager decides to give both E_1 and E_2 the values p and q so that they can use the CRT to decrypt.

i. Does this make the system as fast as normal RSA? Explain your answer.

ii. Why does this defeat the purpose of the system?

Use of the CRT speeds up decryption by a factor of around 4 times. The system is now almost as fast as plain RSA since the decryptions can be run in parallel. But either party can recover the whole of d and decrypt alone.

Alternate answer: The system is now almost half as fast since the decryptions cannot be run in parallel. But either party can recover the whole of d and decrypt alone.

5. Consider the following protocol with the goal of key establishment. This is a repaired version of the Needham–Schroeder protocol.

Here N_A is a nonce chosen by party A, N_B is a nonce chosen by B, and K_{AB} is the session key chosen by server S. ID_A and ID_B are identity strings for A and B respectively. K_{AS} and K_{BS} are key-encrypting keys initially shared between S and A, and between S and B respectively. The notation $\{X\}_K$ denotes authenticated encryption of X with key K.

1. $A \to B : ID_A, N_A$
2. $B \to S : ID_A, ID_B, N_A, N_B$
3. $S \to B : \{N_A, ID_A, ID_B, K_{AB}\}_{K_{AS}}, \{N_B, ID_A, ID_B, K_{AB}\}_{K_{BS}}$
4. $B \to A : \{N_A, ID_A, ID_B, K_{AB}\}_{K_{AS}}$

(a) On receipt of message 4, A should check that the received N_A is the same value as that chosen in message 1. Describe an attack on the protocol if A does not perform this check, including the messages which an attacker sends. What is the consequence of this attack?

Attacker can replay old messages and this would not be detected by A. The attacker can capture $\{N_A, ID_A, ID_B, K'_{AB}\}_{K_{AS}}$ from a previous protocol run, obtain the old session key K'_{AB}, and then it will be accepted by A.

(b) On receipt of message 3, B should also check that the received ID_A is the same identity received in message 1 and sent in message 2. Describe an attack if B does not perform this check.

Attacker can change the identity in message 2 so that B shares the key with, say, C instead of A.

Alternate question:

(b) Suppose that instead of using authenticated encryption, plain encryption by a synchronous stream cipher is used, such as AES in counter mode. How does this also allow an attack?

Attacker can change any known field. For example, ID_B can be changed to ID_C in message 4 by adding in $ID_B \oplus ID_C$. This will trick A to think the key is shared with C instead of B.

6. The Signal messaging protocol uses two kinds of *ratcheting* to update the keys used to protect messages: Diffie–Hellman ratcheting is used when the next message is sent in the opposite direction from the previous message; symmetric ratcheting with a hash function is used when the next message is sent in the same direction. Assume a powerful adversary who can capture and delay messages and has the ability to compromise devices later.

(a) How does the ratcheting in Signal improve the security of messages against this adversary, in comparison to the security of:

i. email messages encrypted with PGP;
There is no forward secrecy for PGP encryption, so the adversary will obtain all previous messages.

ii. messages sent as application data in a TLS 1.3 session.

The main difference is that a different key is used to protect each message. If a device is compromised, only the key to protect the current message will be revealed. Older messages cannot be found.

(b) If several messages are sent in the same direction in the Signal protocol, how does their security compare to the security of messages sent successively in opposite directions?

An adversary who delays messages and compromises the recipient can obtain all of the messages sent in the same direction. The adversary can only obtain the messages obtained after the compromise since the ratchet is one-way. There is also a (system imposed) limit on how many consecutive messages you can send before you are forced to do a DH-ratchet up and down.