TTM4135 Spring exam 2020: 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 n is any odd number. Then $2^{-1} \bmod n$ is:

   (a) $n-2$
   (b) $(n+1)/2$ ✓
   (c) $(n-1)/2$

   Explanation: $2 \cdot (n+1)/2 \bmod n = n+1 \bmod n = 1$.

2. Suppose a plaintext comes from a natural language, similar to English, where the most frequent character appears with probability 1/10. For which of the following ciphers is the frequency of all ciphertext characters likely to be less than 1 in 10?

   (a) The random simple substitution cipher
   (b) A transposition cipher on blocks of size 12
   (c) The Vigenére cipher with a key of length 8 ✓

   Explanation: The Vigenére cipher smooths the frequency of the plaintext characters so we expect that the most frequent plaintext character will be split into ciphertext characters of lower frequencies.

3. Which of the following ciphers has the largest number of possible keys?

   (a) The triple-DES cipher with two independent keys
   (b) The random simple substitution cipher with alphabet consisting of all 8-bit bytes ✓
   (c) The AES block cipher with the shortest allowed key length

   Explanation: The alphabet for the simple substitution has 256 characters so there are 256! keys. A very crude estimate is that $256! >> 128^{128} = 2^{896}$ which is much more than the other two ciphers (or any modern standard cipher).

4. A block cipher is a function $E(m,k)$ which takes a plaintext block m and a key k to a ciphertext block c. In any useful block cipher:

   (a) for a fixed key k, the function $E(\cdot,k)$ is a permutation of the set of plaintext blocks ✓
   (b) for a fixed plaintext block m, the function $E(m,\cdot)$ is a permutation of the set of keys
   (c) for a fixed ciphertext block c, there is a unique pair (m,k) such that $E(m,k)=c$

Explanation: For a fixed k, there must be exactly one ciphertext block which is the encryption of any plaintext block, otherwise it would not be possible to decrypt.

5. Suppose the string `ABCDE` is a portion of a ciphertext which has been encrypted with the one time pad. The corresponding plaintext string P is another 5-character string. Which of the following is the most accurate statement?

   (a) Every possible plaintext string of length 5 is equally likely to be the correct P

   (b) The attacker gains nothing useful about the correct P from seeing `ABCDE` ✓

   (c) The correct P is the same plaintext corresponding to any previous portion of ciphertext equal to `ABCDE`

   Explanation: Although in general the plaintexts are not equally likely, the ciphertext from the one time pad does not help an attacker in guessing what was the correct plaintext.

6. The discrete logarithm problem in $\mathbb{Z}_p^*$ is the basis for many public key cryptosystems. On conventional computers (not quantum computers) there is:

   (a) no known polynomial time algorithm ✓

   (b) no known subexponential time algorithm

   (c) no known exponential time algorithm

   Explanation: The number field sieve is a sub-exponential time algorithm for the DLP in $\mathbb{Z}_p^*$.

7. The security of RSA encryption is related to the problem of integer factorisation in the following way:

   (a) if RSA cannot be broken then factorisation is hard ✓

   (b) if RSA can be broken then factorisation is easy

   (c) If RSA is secure then factorisation may be easy or hard

   Explanation: If factoring is easy then an attacker can find the RSA public key from the private key, so that RSA is definitely broken. This is the same as saying that if RSA cannot be broken than factorisation must be hard.

8. The RSA encryption algorithm uses a public exponent e, a private exponent d, and a public modulus n. In order to speed up the decryption process, it is common to:

   (a) choose a small value for e

   (b) choose a small value for d

   (c) apply the Chinese Remainder theorem ✓

Explanation: The owner of the private decryption key also can have access to p and q, allowing application of the CRT

9. ElGamal encryption works in $\mathbb{Z}_p^*$. The ciphertext of a message m is a pair (g^k, my^k). To avoid a known plaintext attack it is essential to:

   (a) choose a new k for each encryption ✓

   (b) choose a new y for each encryption

   (c) choose a new g for each encryption

   Explanation: Since y is the public key of the recipient, a new k must be chosen so that y^k is different for each message. If not then y^k can be extracted from one known plaintext/ciphertext pair and used to decrypt other messages.

10. Let h be the identity function, $h(x) = x$ defined on bit strings of length 128-bits. This function does not meet the property of being:

    (a) oneway ✓

    (b) collision-resistant

    (c) second pre-image resistant

    Explanation: Given an output value y we know that the input to h was also y, thus h is not one-way.

11. HMAC is a takes as input a key K and message M and outputs a tag T. Suppose that HMAC uses the hash function is SHA-256. If $HMAC(K, M_i)$ is computed for many different messages M_i (and a fixed K) then two identical tags will probably first appear after:

    (a) 2^{16} tag values are computed

    (b) 2^{128} tag values are computed ✓

    (c) 2^{255} tag values are computed

    Explanation: By the birthday paradox, collisions are likely to appear after around the square root of the number of possible tags have been computed, so around $2^{256/2}$.

12. A difference between a message authentication code (MAC) and a digital signature scheme is:

    (a) a signature must be randomised but a MAC tag need not be

    (b) a MAC tag can be recomputed by the verifier but a signature cannot be ✓

    (c) a MAC provides data integrity but a signature does not

Explanation: The recipient of a MAC shares the key with the sender and can recompute to verify the tag; for a signature only the private key owner can compute signatures.

13. The Kerberos protocol makes use of a ticket containing four values $(K_{AB}, ID_A, ID_S, N_A)$ which is shared between a client A and server S. A suitable algorithm to use to protect these values would be:

    (a) AES in GCM mode ✓

    (b) AES in CBC mode

    (c) HMAC

    Explanation: The ticket needs to be confidential, to hide K_{AB}, and preserve integrity of the identities and nonce. Therefore AES in GCM mode is the only option.

14. A difference between TLS 1.3 and TLS 1.2 is:

    (a) the TLS 1.3 handshake protocol always provides forward secrecy ✓

    (b) there are no known attacks on the TLS 1.3 protocol

    (c) the TLS 1.3 record protocol includes data compression

    Explanation: Only Diffie-Hellman handshakes are allowed in TLS 1.3 since the RSA method in TLS 1.2 was removed.

15. Many email servers add a DomainKeys Identified Mail (DKIM) digital signature to outgoing mail. This signature:

    (a) can be verified by any recipient of the email ✓

    (b) is verified and then removed by the receiving domain mail server

    (c) can only be verified by the receiving domain

    Explanation: The signature is openly available in the mail headers and can be verified by anyone using the public key which can be obtained from the DNS record.

2 Written answer questions

1. Consider a variant of the Hill cipher which has the encryption equation

$$C = KP + L \bmod n$$

where the key has two parts, a 2×2 matrix K and 2×1 column vector L. The column vectors C and P represent the ciphertext and plaintext respectively. Here n is the size of the alphabet in use. In this question all matrices are 2×2. If $L = 0$ then this variant is the same as the basic Hill cipher.

   (a) What is the decryption equation for this variant cipher?
   $P = K^{-1}(C - L) \bmod n$ (1 mark)
   Note that matrix multiplication is not commutative, so it matters in which order the computation is done.

   (b) What is the possible number of keys in this variant? Write an expression in terms of n and the number of keys of the basic Hill cipher.
   $n^2 \times N_{BH}$ where N_{BH} is the number of keys in the basic Hill. Not all matrices are valid for the basic Hill, so it is inaccurate to say that $N_{BH} = n^4$. (1 mark)

   (c) Explain how an attacker can use a chosen plaintext attack to obtain the key with three chosen plaintext pairs.
   Suppose that (P_1, P_2, P_3) are three plaintext column vectors with corresponding ciphertext vectors (C_1, C_2, C_3). Then

$$C_1 = K \cdot P_1 + L \tag{1}$$
$$C_2 = K \cdot P_2 + L \tag{2}$$
$$C_3 = K \cdot P_3 + L \tag{3}$$

   By setting $P_1 = 0$ (column vector) we can first find L since then $C_1 = L$. Then we can subtract L to get a square matrix equation: $((C_2 - L)(C_3 - L)) = K \cdot (P_2 P_3)$ which can be solved as the basic Hill for K by inverting $(P_2 P_3)$. Note that in a chosen plaintext attack P_2 and P_3 can be chosen to ensure that $(P_2 P_3)$ is invertible. (3 marks)

2. A non-standard mode of operation for block ciphers has the following general equation for computing each output block: $C_t = E(P_t \oplus P_{t-1} \oplus C_{t-1}, K)$ where $C_0 = IV$ which is sent with the ciphertext and $P_0 = 0$ (the block of all 0 bits).

   Alternate question:
   $C_t = E(P_t \oplus C_{t-1}, K) \oplus P_{t-1}$

   (a) What is the equation for decryption of ciphertext block C_t to obtain the plaintext block P_t?

$$P_t = D(C_t, K) \oplus P_{t-1} \oplus C_{t-1}$$

   (1 mark)
   Alternate answer

$$P_t = D(C_t \oplus P_{t-1}, K) \oplus C_{t-1}$$

(b) Suppose that there is an error in transmission when block C_t is sent to a recipient, so that one bit is changed. How many blocks, or partial blocks, are changed when the receiver decrypts? Explain your answer.

One bit change in C_t will give a random output for $D(C_t, K)$. So P_t will be random. But this randomness will also then propagate to P_{t+1} since P_t gets added in. Thus all subsequent plaintext blocks get changed randomly (but not independently). (2 marks)

(c) Is it possible to encrypt multiple plaintext blocks in parallel? Is it possible to decrypt multiple blocks in parallel? Explain your answers.

To encrypt in parallel you need to have the previous C_{t-1} block so parallel encryption is not possible.
Parallel decryption is possible by first finding all the $D(C_t, K)$ values and then adding the previous P_{t-1} and C_{t-1} blocks. *(However, the statement that parallel decryption is not possible was also accepted.)* (2 marks)

Alternate answer:
To encrypt in parallel you need to have the previous C_{t-1} block so parallel encryption is not possible.
To decrypt in parallel you need to have the previous P_{t-1} block so parallel decryption is not possible.

3. Two efficient tests for primality of an integer n are the Fermat test and the Miller–Rabin test. Both tests use a base value a chosen randomly in the range $1 < a < n - 1$ and are usually run for multiple bases. Suppose that the tests are being used to test $n = 45$.

Alternate question: Numbers are $n = 85$ and $a = 4$

(a) Show that $19^2 \bmod 45 = 1$.

$19 \times 19 \bmod 45 = 361 \bmod 45 = 8 \times 45 + 1 \bmod 45 = 1$. (1 mark)
Alternate answer: $16 \times 16 \bmod 85 = 256 \bmod 85 = 1$.

(b) Show that the Fermat test will return that $n = 45$ is a probable prime if the value $a = 8$ is chosen.

$8^2 \bmod 45 = 19$. Thus from part (a), $8^4 \bmod 45 = 19^2 \bmod 45 = 1$
$8^{44} \bmod 45 = (8^4)^{11} \bmod 45 = 1$ (2 marks)

Alternate answer:
$4^2 \bmod 85 = 19$. Thus from part (a), $4^4 \bmod 85 = 16^2 \bmod 85 = 1$
$4^{84} \bmod 85 = (4^4)^{21} \bmod 85 = 1$

(c) Show that the Miller-Rabin test will return that $n = 45$ is composite if the value $a = 4$ is chosen.

$44 = 11 \times 4$. First compute
$8^{11} \bmod 45 = 8^8 \times 8^3 = 1 \times 19 \times 8 \bmod 45 = 152 \bmod 45 = 17$
$17^2 \bmod 45 = 19$
$19^2 \bmod 45 = 1$ (from part (a))
Since -1 never occurs in the sequence the Miller–Rabin test returns composite. (2 marks)

Alternate answer:
$84 = 21 \times 4$. First compute
$4^{21} \bmod 85 = 4^{20} \times 4 = 1 \times 4 \bmod 85 = 4$
$4^2 \bmod 85 = 16$
$16^2 \bmod 85 = 1$ (from part (a))
Since -1 never occurs in the sequence the Miller–Rabin test returns composite.

4. Consider the Diffie–Hellman protocol in the group $\mathbb{Z}_p^*$. In order to add authentication to the basic Diffie–Hellman protocol it is common to use digital signatures. An alternative is to use a long-term key directly in the protocol. Suppose that A has long-term secret key x with public key g^x and B has long-term secret key y with public key g^y. The protocol is then as follows.

   - A chooses random a and sends the value $A = g^a$ to B.
   - B chooses random b and sends the value $B = g^b$ to A.
   - A computes the session key $K_{AB} = B^x(g^y)^a$, using the received message Y and the long-term key of B.
   - B computes the session key $K_{BA} = A^y(g^x)^b$, using the received message X and the long-term key of A.

   (a) Show that A and B compute the same key: $K_{AB} = K_{BA}$.

$$\begin{aligned} K_{AB} &= B^x(g^y)^a \\ &= g^{bx}g^{ay} \\ &= g^{bx+ay} \end{aligned}$$

   By symmetry this is also equal to K_{BA}. (2 marks)

   (b) Show that this protocol does not achieve the forward secrecy property.

   We suppose that x and y are compromised after the session under attack is completed. Then the attacker has recorded A and B can compute:

$$\begin{aligned} K_{AB} &= A^x B^y \\ &= g^{ax+by} \end{aligned}$$

   (3 marks)

5. Two signature schemes commonly used today are RSA signatures and DSA signatures. Suppose that the RSA modulus n has length 2048 bits and the DSA modulus p has length 2048 bits with length of parameter q equal to 224 bits. You may assume that the DSA parameters p and q are fixed for all parties.

   Alternate question: Parameters are $|n| = 3072$, $|p| = 3072$, $|q| = 256$.

   (a) RSA signatures often use a public exponent $e = 2^{16} + 1$. Approximately how much faster on average is signature verification with this value of e compared to when e is randomly chosen e?

   With short exponent need only 17 multiplications. With random exponent around 3072. So ratio is around 192:1 (or 180 is more precise). (2 marks)

   Alternate answer: With short exponent need only 17 multiplications. With random exponent around 4608. So ratio is around 288:1 (or 271 is more precise).

   (b) What is the approximate ratio of the signature length, RSA against DSA?

   DSA has two components each size q, so 448 bits while RSA uses 2048 bits. So ratio is about 1:4.5. (1 mark)

   Alternate answer: DSA has two components each size q, so 512 bits while RSA uses 3072 bits. So ratio is about 1:6.

(c) What is the approximate ratio of the signing key length, RSA against DSA?

RSA key has size 2048 for private exponent. DSA has short exponent of 224 bits. So 9:1 approximately. *Also acceptable to include RSA modulus, when ratio becomes 18:1.* (1 mark)

Alternate answer: RSA key has size 3072 for private exponent. DSA has short exponent of 256 bits. So 12:1 approximately. *Also acceptable to include RSA modulus, when ratio becomes 24:1.*

(d) What is the approximate ratio of the verification key length, RSA against DSA?

RSA key has size 2048 for modulus, but could have fixed exponent. DSA has public key of 2048 bits. So 1:1 approximately. (1 mark)

Alternate answer: RSA key has size 3072 for modulus, but could have fixed exponent. DSA has public key of 3072 bits. So 1:1 approximately.

6. Protection of metadata, typically included in header information in network protocols, is important in preserving privacy.

(a) Compare what cryptographic protection is available for IP metadata in (i) IPSec in tunnel mode and (ii) TLS between a client and server.

In tunnel model IPSec encapsulates IP header information so that it is hidden from eavesdroppers. In contrast, TLS in normal usage does not touch IP headers since it works at the top of the transport layer (2 marks)

(b) Compare what cryptographic protection is available for email metadata in (i) PGP and (ii) STARTTLS.

PGP protects mail contents end-to-end without changing the headers which are necessary to deliver the mail. In contrast, STARTTLS works between mail servers and encapsulates the mail headers, but this protection relies on trust in the mail servers. (2 marks)

(c) Is there a conflict between protection of metadata and end-to-end security? Discuss such a conflict in the context of the above two examples.

IPSec in tunnel model and STARTTLS are not end-to-end and both encapsulate their payload to hide the metadata. However, end-to-end communication needs to use the header information, as with PGP and TLS. A solution is to use both. (1 mark)