Question 1.

See tables in IS.

Question 2.

Using *Chinese remainder theorem*, which says that for our system of congruences:

$$\begin{aligned} x &\equiv a_1 \ (mod\ m_1) & \rightsquigarrow && x &\equiv 8 \ (mod\ 17) \\ x &\equiv a_2 \ (mod\ m_2) & \rightsquigarrow && x &\equiv 4 \ (mod\ 19) \\ x &\equiv a_3 \ (mod\ m_3) & \rightsquigarrow && x &\equiv 19 \ (mod\ 23) \end{aligned}$$

there is one unique solution:

$x = a_1 b_1 b_1^{-1} + a_2 b_2 b_2^{-1} + a_3 b_3 b_3^{-1} \ (mod\ m_1 m_2 m_3)$, where $b_k = \dfrac{m_1 m_2 m_3}{m_k} \wedge b_k^{-1}$ is modular inverse of b_k.

We also know that $xy \equiv xz \ (mod\ n) \Rightarrow y \equiv z \ (mod\ n)$. Thus:

$$\begin{aligned} b_1 &= 19 * 23 = 437 \rightsquigarrow 437 b_1^{-1} \equiv 1 \ (mod\ 17) & \rightsquigarrow && 12 b_1^{-1} &\equiv 1 \ (mod\ 17) \rightsquigarrow b_1^{-1} = 10 \quad (\star) \\ b_2 &= 17 * 23 = 391 \rightsquigarrow 391 b_2^{-1} \equiv 1 \ (mod\ 19) & \rightsquigarrow && 11 b_2^{-1} &\equiv 1 \ (mod\ 19) \rightsquigarrow b_2^{-1} = 7 \quad (\star) \\ b_3 &= 17 * 19 = 323 \rightsquigarrow 323 b_3^{-1} \equiv 1 \ (mod\ 23) & \rightsquigarrow && b_3^{-1} &\equiv 1 \ (mod\ 23) \rightsquigarrow b_3^{-1} = 1 \end{aligned}$$

Then $x = 8 * 437 * 10 + 4 * 391 * 7 + 19 * 323 \ (mod\ 7429) = 52045 \ (mod\ 7429) = \mathbf{42}$.
$(\star)$ - find these as *Bézout coefficients* using *Extended Euclidian algorithm*

Or we could just write them out and see:

$$8 + 17k_1 \rightsquigarrow 8, 25, 42, ... \quad 4 + 19k_2 \rightsquigarrow 4, 23, 42, ... \quad 19 + 23k_3 \rightsquigarrow 19, 42, ...$$

Question 3.

(a) $K_{AB} = g_A(r_A, s_B)$
$= a_A * r_B + b_A * s_B$
$= ((a * r_A) + (b * s_A)) * r_B + ((b * r_A) + (c * s_A)) * s_B$
$= (a * r_A * r_B) + (b * s_A * r_B) + (b * r_A * s_B) + (c * s_A * s_B)$
$= ((a * r_B) + (b * s_B)) * r_A + ((b * r_B) + (c * s_B)) * s_A$
$= (a_B * r_A) + (b_B * s_A)$
$= g_B(r_A, s_A))$
$= K_{BA}$

(b) In my opinion is this protocol less secure than the original protocol.
$a_U = (a + b * r_U)$ in the original protocol, we can also see it as $a_U = (a * 1 + b * r_U)$ or $a_U = (a * (s_U = 1) + b * r_U)$ it means that $gcd(s_U, r_U) = 1$
In this version $a_U = (a * r_U + b * s_U)$, so s_U and r_U are swapped and s_U is not only 1, but some other number $< p$.
I would say that the threat is when $gcd(s_U, r_U) \neq 1$ as a_U, b_U and also the key could be divided by the gcd, which is a security issue.

Question 4.

(a) We know that $p \equiv 3 \mod 4$ and $q \equiv 3 \mod 4$: therefore, $p \equiv 3 \mod 8$ or $p \equiv 7 \mod 8$ and $q \equiv 3 \mod 8$ or $q \equiv 7 \mod 8$. Since $p \neq \pm q \mod 8$, if $p \equiv 3 \mod 8$, then $q \equiv 7 \mod 8$, and vice-versa.

In our case, $N = p \times q$, then by definition, $N \equiv 3 \times 7 \mod 8 \equiv 21 \mod 8 \leftrightarrow N \equiv 5$.

As given here (https://en.wikipedia.org/wiki/Jacobi_symbol) in the statement 8 of the section "properties, we have $\left(\frac{2}{n}\right) = (-1)^{\frac{n^2-1}{8}} = \begin{cases} 1 & \text{if } n \equiv 1, 7 \pmod 8, \\ -1 & \text{if } n \equiv 3, 5 \pmod 8). \end{cases}$.

Since, in our case, $N \equiv 5 \mod 8$, we can deduce that $\left(\frac{2}{n}\right) = -1$.

(b) The Jacobi symbols for x, $N - x$, $2x$ and $N - 2x$ are respectively $\left(\frac{x}{N}\right)$, $\left(\frac{N-x}{N}\right)$, $\left(\frac{2x}{N}\right)$ and $\left(\frac{N-2x}{N}\right)$. We can rewrite some of them:

$$\left(\frac{N-x}{N}\right) = \left(\frac{-x+N}{N}\right) = \left(\frac{-x}{N}\right) = \left(\frac{x}{N}\right) \textit{ since } N \equiv 1 \mod 4$$

$$\left(\frac{2x}{N}\right) = \left(\frac{2}{N}\right)\left(\frac{x}{N}\right) = -1 \times \left(\frac{x}{N}\right)$$

$$\left(\frac{N-2x}{N}\right) = \left(\frac{-2x+N}{N}\right) = \left(\frac{-2x}{N}\right) = \left(\frac{2}{N}\right)\left(\frac{-x}{N}\right) = -1 \times \left(\frac{x}{N}\right) \textit{ since } N \equiv 1 \mod 4$$

Therefore, if $\left(\frac{x}{N}\right) = 1$, then $\left(\frac{N-x}{N}\right) = 1$, and on the contrary $\left(\frac{2x}{N}\right) = -1$ and $\left(\frac{N-2x}{N}\right) = -1$ (and vice-versa). In such a case, neither $2x$ nor $N - 2x$ are square modulo N.

Let us suppose that, for a given value of x, $\left(\frac{x}{N}\right) = 1$ (the demonstration is similar in the opposite case). This does not guarantee that x is a square modulo N because N is not a prime. We must decompose our symbols $\left(\frac{x}{N}\right)$ and $\left(\frac{N-x}{N}\right)$:

$$\left(\frac{x}{N}\right) = \left(\frac{x}{p}\right)\left(\frac{x}{q}\right)$$

$$\left(\frac{N-x}{N}\right) = \left(\frac{N-x}{p}\right)\left(\frac{N-x}{q}\right) = \left(\frac{-x+qp}{p}\right)\left(\frac{-x+pq}{q}\right) = \left(\frac{-x}{p}\right)\left(\frac{-x}{q}\right)$$

If $\left(\frac{x}{p}\right) = 1$ AND $\left(\frac{x}{q}\right) = 1$, since p and q are primes, then x is a square modulo p and modulo q, which implies that it is a square modulo N. However, if x is a square modulo N, then $-x$ is not.

As we can see, exactly 2 numbers among x, $N-x$, $2x$ and $N-2x$ have Jacobi symbols equal to 1: those who have not are not squares. Between the two numbers with a Jacobi number equal to 1, only one of them is actually a square modulo N. This is the proof that, $\forall 1 \leq x < N$, exactly one among x, $N - x$, $2x$ and $N - 2x$ is a square modulo N.

Question 5

(*6 points*, 4+2) Consider a cryptosystem where an intended recipient performs the following:

- Chooses n numbers x_i with $\gcd(x_i, x_j) = 1$, $i \neq j$.
- Chooses a prime number q such that
$$q \geq \prod_{i=0}^{n} x_i.$$
- Chooses a primitive root b modulo q.
- Calculates a_i, $1 \leq i \leq n$ such that
$$x_i \equiv b_i^a \pmod{q}.$$
- The values a_i, $1 \leq i \leq n$ form the public key; q and b_i, $1 \leq i \leq n$ remain secret.

To send an n-bit message $(m_1, \ldots, m_n)$ where $m_i \in \{0, 1\}$, the sender calculates
$$k = \sum_{i=1}^{n} m_i a_i$$
and sends k to the recipient.

(a) How does the intended recipient recover the message? Explain.

(b) The security of this cryptosystem relies on which assumptions?

Solution This is the Merkle-Hellman multiplicative version of knapsack.

(a) The recipient calculates
$$m \equiv b^k \pmod{q}.$$
Since
$$b^k \equiv \prod_{i=1}^{n} (b^{a_i})^{m_i} \equiv \prod_{i=1}^{n} x_i^{m_i} \pmod{q}$$
and $q \geq \prod_{i=0}^{n} x_i$ then
$$m = \prod_{i=0}^{n} x_i^{m_i}$$
and $m_i = 1$ iff $x_i | m$.

(b) Discrete logarithm problem and knapsack problem.

Question 6.

(a) It is always an one-way function. If we add arbitrary padding such as $0 \ldots 0$ to the output of a one-way function it does not affect its one-wayness because the zeros can be removed and the problem is same as solving the original preimage problem. If we duplicate the same one-way function we can split the encoded string to two parts and again obtain the same preimage problem as if we were solving the original.

(b) It is not always a one-way function. Let's have a one-way function f that maps binary strings of length n to another binary strings of length n. We can construct a one-way function g that maps the binary strings of length n to binary strings of length $2n$ by using the output of one-way function f and adding padding of n zeros. This is still a one-way function because if we remove the padding zeros it is the same problem as finding preimage of the f. Now let's create another one-way function h that maps binary strings of length $2n$ to binary strings of length $2n$. It returns $2n$ zeros if the first half of the string is all zeros and result of g other wise. We can clearly see that if we call the function h once on a input that has at least one non-zero digit in the first half it will return the result of the function g that returns the message with first half zeros and if we use that as an input of h the second application of that function we will obtain all zero result.

Question 7.

Eve might be capable of decrypting the original message m.

In the RSA, n and e are the public key. We also know that $2^{511} < n \leq 2^{512}$ and $e = 3$.

Bob chunks the message into 64-bit long parts, which means $2^{64^3} = 2^{192}$ values for the cipher message. This is considerably less than the modulus which is at minimum $2^{511} + 1$. Therefore the modulo operation is never used and $c_i = m_i^{e=3}$, so Eve could decrypt all chunks sent simply by computing $m_i = \sqrt[e=3]{c_i}$. The only thing Eve has to manage is to identify all these chunks, but these are separated with the unique identifier # and therefore she can spot this recurrence and identify these chunks. Then she has to try to compute the root as shown above.