Exercise 4

Question 1:

a) $\sum_{i=1}^{t} a_i = h$; where $a_i \in \mathbb{Z}_n$

Requirement	Explanation	Satisfied
• Variable input size	The function can be applied to any sequence of length t	Yes
• Fixed output size	The output grows at the same time as t	No
• Preimage resistance	Given the sum h, it is easy to find the string that created it	~~Yes~~ No
• Efficiency	The sum h is easy to calculate	Yes
• Second preimage resistance	You can easily construct sequences with the sum	No
• Collision resistance	Sum sequences have the same output	No
• Pseudorandomness	The output is not random	Yes

In conclusion, the function $h = \sum_{i=1}^{t} a_i$ is not cryptographically secure.

b) $h = \left(\sum_{i=1}^{t} (a_i)^2 \right) \bmod n$; where $a_i \in \mathbb{Z}_n$

Requirement	Explanation	Satisfied
Variable input size	The function can be applied for any t (length message)	Yes
Fixe output size	The output will be always at $1 \ldots n$	Yes
Efficency	Easy operations	Yes

Preimage resistance	Since there are easy operations we can just try values until they match	No
Second preimage resistance	Different sequences can produce the same of squares mod n	No
Collision resistance	High chance, collisions are easy to be made	No
Pseudorandomness	the output is never random	No

In conclusion the function is not cryptographically secure.

c) $M = (189, 632, 900, 712, 347)$; $n = 989$

$\lambda = (189^2 + 632^2 + 900^2 + 712^2 + 347^2) \bmod 989$

$\lambda = 1,888,230 \bmod 989$

$\boxed{\lambda = 409}$

Question 2

Question 2

a) Bob checks: $M = S + k \cdot y_A$

$S + k \cdot y_A = M - k \cdot x_A G + k \cdot x_A G = M$ => the verification process produces an equality if the signature is valid

b) $M = S + k y_A$ $\quad y_A = x_A G$

we can do

=> $S = M - k y_A$ so there can be another way to compute the signature.

So an attacker can:

1. pick a random key k
2. Calculate $S = M - k y_A$
3. Send to Bob the pair M, k, S

Bob will check $M = S + k y_A \Rightarrow (M - k y_A) + k y_A = M$ which verifies even though we never have the private key x_A

In conclusion, the scheme is unacceptable because the attacker can falsify the signatures and still send Bob a message that verifies.

Question 3

$y^2 = x^3 + 1 \pmod{17}$; $P(15, 13)$

a) $2P = P + P$

$P = (x_1, y_1) = (15, 13) \Rightarrow \lambda = \frac{3x_1^2}{2y_1} \bmod p$

$3x_1^2 = 3 \cdot 15^2 = 675$

$2y_1 = 2 \cdot 13 = 24$

$\Rightarrow \lambda = \frac{675}{36} \bmod 17 = \frac{12}{9} \bmod 17$

$9^{-1} \bmod 17 = x \iff 9x \equiv 1 \bmod 17$ (modular inverse)

$9 \cdot 2 = 18 \equiv 1 \bmod 17 \Rightarrow 9^{-1} = 2 \bmod 17$

$\Rightarrow \lambda = 12 \cdot 2 = 24 \bmod 17 = 7$

$2P = (x_3, y_3)$:

$x_3 = \lambda^2 - 2x_1 = 7^2 - 2 \cdot 15 = 49 - 30 = 19 \bmod 17 = 2$

$y_3 = \lambda(x_1 - x_3) - y_1 = 7(15 - 2) - 13 = 7 \cdot 13 - 13 = 78 \bmod 17 = 0$

$\Rightarrow \boxed{2P = (2, 10)}$

$3P = 2P + P$; $P = (15, 13)$, $2P = (2, 10)$

$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \bmod p = \frac{10 - 13}{2 - 15} = \frac{-3}{-13} \bmod 17 = \frac{14}{4} \bmod 17$

We will find $4^{-1} \bmod 17$

We will try small values: $4 \cdot 13 = 52 \equiv 1 \bmod 17 \Rightarrow 4^{-1} = 13$

$\Rightarrow \lambda = 14 \cdot 13 = 182 \bmod 17 = 182 - 170 = 12$

Finally we compute: $x_3 = \lambda^2 - x_1 - x_2 = 144 - 17 \equiv 127 \bmod 17$

$\Rightarrow \boxed{x_3 = 8}$

$y_3 = \lambda(x_1 - x_3) - y_1 = 12 \cdot (15 - 8) - 13 = 12 \cdot 7 - 13 = 84 - 13 = 71 \bmod$

$\Rightarrow \boxed{y_3 = 3}$

So : $2P = (2, 10)$ and $3P = (8, 3)$

b) The points provided have some duplicates: $(10, 2)$; $(12, 1)$; There are 14 unique points in the group. So the order of the eliptic curve group is 14.

The Eliptic curve cryptography is based on a public-key cryptosystem based on mathematics of eliptic curves over finite fields

As an exaple we will use the curve $y^2 = x^3 + 7 \bmod 17$ with a base point $G = (15, 13)$. Here we Alice and Bob ~~Alice will take~~ as for every cryptology system.
Alice will have private key $a = 5$ and the public key $A = 5G$
Bob will also private key $b = 7$ and the public key $B = 7G$
Now Alice and Bob exchange A and B, this computes:
⇒ ~~Alice~~ Alice will get the encrypted message:

$$S = a \cdot B = 5 \cdot 7G = 35G$$

⇒ Bob will get the encrypted message:

$$S = b \cdot A = 7 \cdot 5G = 35G$$

~~So now they~~

So they have both shared the same secret point S and not having to reveal their private keys.
This is how the Elliptic curve cryptography with an example.