Question 1.

(a) Number we want to encrypt must be lower then n, therefore we can divide $w = 445620$ into two parts, $w_1 = 445$ and $w_2 = 620$. Now we can encrypt the plaintext using $c = w^e \bmod n$ formula.

$$c_1 = 445^7 \bmod 1147 \equiv 75$$
$$c_2 = 620^7 \bmod 1147 \equiv 465$$

For decryption we have to find d. For that we can use extended euclidian algorithm $gcd(\varphi(n), e)$ where $\varphi(1147) = (31-1)(37-1) = 1080$.

$$1080 = 154 \cdot 7 + 2$$
$$7 = 3 \cdot 2 + 1$$

$$1 = 7 - 3 \cdot 2 = 7 - 3(1080 - 154 \cdot 7) = -3 \cdot 1080 + 463 \cdot 7$$

Here we found an inverse of e which is $d = 463$. Now we can decrypt c_1 and c_2 back into plaintext using $w = c^d \bmod n$ formula.

$$w_1 = 75^{463} \bmod 1147 \equiv 445$$
$$w_2 = 465^{463} \bmod 1147 \equiv 620$$

And we obtained the correct plaintext $w = 445620$.

(b) We encrypt plaintext $w = 0010100$ using formula $c = X'w^\top$

$$c = X'w^\top = \begin{pmatrix} 155 & 208 & 57 & 216 & 126 & 150 & 153 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = 183$$

For decryption we first need to compute $u^{-1} \bmod m$ which we can accomplish using extended euclidian algorithm $gcd(m, u)$.

$$257 = 1 \cdot 155 + 102$$
$$155 = 1 \cdot 102 + 53$$
$$102 = 1 \cdot 53 + 49$$
$$53 = 1 \cdot 49 + 4$$
$$49 = 12 \cdot 4 + 1$$

$1 = 49-12\cdot 4 = 49-12(53-49) = -12\cdot 53+13\cdot 49 = -12\cdot 53+13(102-53) = 13\cdot 102-25\cdot 53 =$
$13\cdot 102-25(155-102) = -25\cdot 155+38\cdot 102 = -25\cdot 155+38(257-155) = 38\cdot 257-63\cdot 155$

We obtained $u^{-1} = -63 \equiv 194 \bmod 257$. Now compute the cryptotext c' for the original Knapsack problem using formula $c' = u^{-1}c \bmod m$.

$$c' = 194 \cdot 183 \bmod 257 \equiv 36$$

Now solve the knapsack problem with the super increasing vector X and cryptotext c'.

$$36 > 29 = x_5$$

$$x_5 = 29 > 36 - 29 = 7 = x_3$$

We have the third and fifth bit equal to 1. Therefore $w = 0010100$.

Question 2.

(a) We need to solve the following system of two equations with two variables:

$$10812 = (p-1)(q-1)$$
$$11021 = pq$$

We can do this by expressing p from the first equation $p = pq - 10812 - q + 1 = 11021 - 10812 - q + 1 = 210 - q$. Plugging this into the second equation we get the quadratic equation $q^2 - 210q + 11021 = 0$. The two possible solutions $q_1 = 107$ and $q_2 = 103$ correspond to the fact that p and q are interchangeable and so we obtain the unique factorization $11021 = 107 \cdot 103$.

(b) In order to factor $n = 53916647$, it is enough to test $x > \sqrt{n}$ until x is found such that $x^2 - n = y^2$.

$$\sqrt{n} \doteq 7342,8$$

For $x = 7343$, $x^2 - n$ is not a square.
For $x = 7344$, $x^2 - n = 17689 = y^2$ is a square.
From there $p + q = 2x$ and $p - q = 2y$. Therefore

$$p = x + y = 7344 + 133 = 7477$$
$$q = x - y = 7344 - 133 = 7211$$

Question 3.

Because we know that the corresponding private key was not carefully chosen let's assume that u is x'_1. Due to this assumption we can calculate X by $ux_i = x'_1 \bmod m$, so
$X = (1, 5, 7, 15, 31, 63, 125, 251, 524, 1111)$. When we want to decrypt the ciphertext (3867, 2085, 2688, 5301, 7390), so we need to compute $u^{-1} \bmod m$. To find the inverse we use the extended Euclidean Algorithm and Bezout's identity for u and m, so u^{-1}=67. By multiplying cryptotexts with u^{-1} and modulo m we get new cryptotext C'=(1377, 1635, 618, 813, 415). First we solve the problem for $c'_1 = 1377$.

$$1377 > 1111 = x_{10}$$

$$x_{10} = 1111 > 1377 - 1111 = 266 > 251 = x_8$$

$$266 - 251 = 15 = x_4$$

We have the tenth, eighth and fourth bit equal to 1. Therefore $w_1 = 0001000101$. $c'_2 = 1635$.

$$1635 > 1111 = x_{10}$$

$$x_{10} = 1111 > 1635 - 1111 = 524 = x_9$$

We have the tenth and ninth bit equal to 1. Therefore $w_2 = 0000000011$. $c'_3 = 618$.

$$618 > 524 = x_9$$

$$x_9 = 618 > 618 - 524 = 94 > 63 = x_6$$

$$94 - 63 = 31 = x_5$$

We have the ninth, sixth and fifth bit equal to 1. Therefore $w_3 = 0000110010$. $c'_4 = 813$.

$$813 > 524 = x_9$$

$$x_9 = 813 > 813 - 524 = 289 > 251 = x_8$$

$$x_6 = 63 > 289 - 251 = 38 > 31 = x_5$$

$$x_4 = 15 > 38 - 31 = 7 = x_3$$

We have the ninth, eighth, fifth and third bit equal to 1. Therefore $w_4 = 0010100110$. $c'_5 = 415$.

$$415 > 251 = x_8$$

$$x_8 = 251 > 415 - 251 = 164 > 125 = x_7$$

$$x_6 = 63 > 164 - 125 = 39 > 31 = x_5$$

$$x_4 = 15 > 39 - 31 = 8 > 7 = x_3$$

$$x_2 = 5 > 8 - 7 = 1 = x_1$$

We have the eighth, seventh, fifth and third and first bit equal to 1. Therefore $w_5 = 1010101100$. And, in the binary form, solutions B of equations $XB^\top = c$' have the form (0001000101, 0000000011, 0000110010, 0010100110, 1010101100). In order to decrypt an English cryptotext, we first decode by 5-bit numbers. Therefore, the resulting plaintext is: **BE CAREFUL**.

Question 4.

Since $w^e = c \mod n$, we can see that $(w_1 w_2)^e = w_1^e \times w_2^e = c_1 \times c_2 \mod n$. Thus product of two plaintexts encrypts to product of ciphertexts of the corresponding plaintexts.

Since $321 \times 562 = 323 \mod 1147$ and $321 \times 1081 \times 562 = 475 \mod 1147$, we conclude that the plaintext for 323 is $21 \times 33 \mod 1147 = 693$, and the plaintext for 475 is $21 \times 29 \times 33 \mod 1147 = 598$.

Question 5.

Since $f(m) = m + \sum_{j=1}^{l} Q_j P_j$ and roots of P_j are also roots of $Q_j P_j$ as $Q_j(...) \times 0 = 0$, Alice performs the decryption by taking the polynomial received from Bob and evaluating it for v. This yields:

$$f(m)(v) = m + \sum_{j=1}^{l} Q_j(v) P_j(v) = m + \sum_{j=1}^{l} Q_j(v) \times 0 = m + \sum_{j=1}^{l} 0 = m$$

The trapdoor here is the valuation v. Finding roots for a system of polynomials is NP-HARD, so the attacker can't find it easily. The randomly chosen polynomials Q are added by Bob so the attacker can't simply subtract P to get the result.

Question 6.

First we express $n = 2^2 \cdot 1749 + 1$, so $s = 2$ and $d = 1749$, $r \in \{0, 1, 2\}$. Now we determine whether the required condition

$$x^d \not\equiv 1 \pmod n \wedge \forall r,\ 0 \le r \le s\colon\ x^{2^r d} \not\equiv -1 \pmod n$$

holds for some x. If it does, n is not a prime.

(i) $x = 2101$

$$2101^{1749} \equiv 6996 \equiv -1 \pmod{6997}$$
$$2101^{2 \cdot 1749} \equiv 1 \pmod{6997} \implies \text{C(2101) does not hold.}$$

(ii) $x = 3035$

$$3035^{1749} \equiv 1 \implies \text{C(3035) does not hold.}$$

(iii) $x = 6101$

$$6101^{1749} \equiv 5201 \pmod{6997} \implies \text{C(6101) does not hold.}$$

(iv) $x = 30$

$$30^{1749} \equiv 1 \pmod{6997} \implies \text{C(6101) does not hold.}$$

Based on the Rabin Miller's Monte Carlo algorithm, 6997 is a prime with probability of error equal to 2^{-4} , that is 6,25%.

5.7 Alice, Bob, and Charlie use the RSA cryptosystem to communicate. Alice has public key $e_A = 29, n = 20453$, Bob's public key is $e_B = 61, n = 20453$, and Charlie has $e_C = 97, n = 20453$. Bob sent the same message m to both Alice and Charlie. Eve intercepted cryptotexts $c_{B\to A} = 3968$ and $c_{B\to C} = 6390$ sent from Bob to Alice and to Charlie, respectively. Can Eve, who is not using any brute force methods, determine the secret message? Omit the fact that the numbers are small. If your answer is yes, determine m. Justify your reasoning.

Solution: We know that

$$m^{29} \equiv 3968 \mod 20453 \tag{1}$$

and

$$m^{97} \equiv 6390 \mod 20453. \tag{2}$$

First of all we calculate $gcd(e_A, e_C)$ by Extended Euclid algorithm

$gcd(29, 97) = gcd(29, 97-3\cdot 29) = gcd(29, 10) = gcd(29-2\cdot 10, 10) = gcd(9, 10) = gcd(9, 10-9) = gcd(9, 1) = 1$

and we find Bezout's coefficients

$$29 \cdot (-10) + 97 \cdot 3 = 1.$$

We can use following identity

$$m \equiv m^{29\cdot -10} \cdot m^{97\cdot 3} \equiv (m^{29})^{-10} \cdot (m^{97})^3 \mod 20453.$$

Now we can use equivalence 1 and 2

$$m \equiv (m^{29})^{-10} \cdot (m^{97})^3 \equiv 3968^{-10} \cdot 6390^3 \equiv 8675^{10} \cdot 6390^3 \equiv 100 \mod 20453.$$

The message is **100**. □

Question 8.

(a)

$$d(X) = \frac{7}{\log_2 127} \approx 1.001619\ldots$$

(b) In a super-increasing vector X the maximum x is the last element, that means its density $d(X) = \frac{n}{log_2 x_n}$. To maximize $d(X)$, we need to minimize the denominator, that is x_n. A super-increasing vector with minimum possible values is the one starting with 1 with the following $x_i, i > 1$ equal to the sum of all previous elements increased by 1. We obtain a super-increasing vector $X = (1, 1+1 = 2, 1+2+1 = 4, 1+2+4 = 8, \ldots)$ (that is $x_i = 2^{i-1}$ for $1 \le i \le n$). We can see that the last and the biggest element $x_n = 2^{n-1}$. After we plug this in the equation for the density of a vector, we obtain

$$d(X) = \frac{n}{log_2 x_n} = \frac{n}{log_2 2^{n-1}} = \frac{n}{n-1}$$

Any other vector X' will have $x'_n \ge x_n$ and therefore $d(X') \le d(X)$.