CS70 Homework 5

Michael Pham

Spring 2023

Contents

1 Equivalent Polynomials

This problem is about polynomials with coefficients in $\mathrm{GF}(p)$ for some prime $p \in \mathbb{N}$. We say that two such polynomials f and g are equivalent if $f(x) = g(x)$ for every $x \in \mathrm{GF}(p)$.

Problem 1.1. Use Fermat's Little Theorem to find a polynomial with degree strictly less than 5 that is equivalent to $f(x) = x^5$ over GF(5); then find a polynomial with degree strictly less than 11 that is equivalent to $g(x) = 4x^{70} + 9x^{11} + 70$ over GF(11).

Solution. Using Fermat's Little Theorem, we know that $x^p \equiv x \pmod{p}$. Then, it follows that $h(x) = x \equiv x^5$ over $\mathrm{GF}(5)$.

Now, to find a polynomial with a degree strictly less than 11 that is equivalent to $g(x) = 4x^{70} + 9x^{11} + 70$ over $\mathrm{GF}(11)$, we observe that by Fermat's Little Theorem:

$$x^{p-1} \equiv 1 \pmod{p}$$

Thus, we see that $4x^{70} = 4(x^{10})^7 \equiv 4(1)^7 \equiv 4$ over $\mathrm{GF}(11)$ using Fermat's Little Theorem. Next, we observe that $9x^{11} \equiv 9x$ over $\mathrm{GF}(11)$. Finally, $70 \equiv 4$ over $\mathrm{GF}(11)$.

From this, we get $i(x) = 9x + 8 \equiv 4x^{70} + 9x^{11} + 70$ over $\mathrm{GF}(11)$. ■

Problem 1.2. In $\mathrm{GF}(p)$, prove that whenever $f(x)$ has degree $\geq p$, it is equivalent to some polynomial $\tilde{f}(x)$ with degree $< p$.

Solution. Suppose that $f(x)$ has degree $d \geq p$. Then, we see that by definition, $f(x) = c_d x^d + c_{d-1}x^{d-1} + \ldots + a_1 x^1 + a_0 x^0$.

By Fermat's Little Theorem, we know that $x^p \equiv x$ over $\mathrm{GF}(p)$. as p is prime. Furthermore, $x^{p-1} \equiv 1$ over $\mathrm{GF}(p)$.

Now, we see that our polynomial $f(x)$ has terms split into two cases:

1. x^d has a degree $d \geq p$.
2. x^d has a degree $d < p$.

In the first case, we observe that any term x^d with a degree $d \geq p$ can be written as $x^{kd+r} = x^{kd}x^r = (x^d)^k x^r = x^k x^r$, where $kd + r = d$, $0 \leq r < p$, and $k \in \mathbb{N}$. Now, if $k \geq p$, we notice that we can keep on repeating this process of rewriting the term until the degree eventually becomes less than p.

For terms where x^d has a degree $d < p$, the degree already satisfies our condition, and thus we don't have to change them.

Thus, by adding all the changed terms which originally had degree $d \geq p$ with the terms which originally had degree $d < p$, we get a new polynomial $\tilde{f}(x)$ whose degree is less than p.

Thus, we observe that whenever $f(x)$ has degree $d \geq p$, it is equivalent to some polynomial $\tilde{f}(x)$ with degree $< p$. ■

2 One Point Interpolation

Suppose we have a polynomial $f(x) = x^k + c_{k-1}x^{k-1} + \cdots + c_2x^2 + c_1x + c_0$.

Problem 2.1. Can we determine $f(x)$ with k points? If so, provide a set of inputs $x_0, x_1, \ldots, x_{k-1}$ such that knowing points $(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_{k-1}, f(x_{k-1}))$ allows us to uniquely determine $f(x)$, and show how $f(x)$ can be determined from such points. If not, provide a proof of why this is not possible.

Solution. Yes, it is possible to determine $f(x)$ using k points. Observe that because we already know the coefficient for the leading term x^k, then we have only $k-1$ unknown coefficients left. Let $g(x)$ denote the function $g(x) = f(x) - x^k$.

Since $g(x)$ has $k-1$ unknown coefficients, we know then that using Lagrange Interpolation, we can construct a unique $g(x)$ using k points. This can be done by using points $x_0 = 0, x_1 = 1, \ldots, x_{k-1} = k-1$, yielding us the values of $f(0), f(1), \ldots, f(k-1)$. Then, we can compute the value of $g(x_i)$ by simply doing $f(x_i) - x^i$ for $i = 0, \ldots, k-1$. And with k points, we can construct a unique polynomial $g(x)$.

Then, from here, we can add $g(x)$ to x^k, yielding us $g(x) + x^k = f(x)$, which is a unique polynomial. Thus, we have uniquely determined $f(x)$ using k points. ■

Problem 2.2. Now, assume each coefficient is an integer satisfying $0 \leq c_i < 100 \quad \forall i \in [0, k-1]$. Can we determine $f(x)$ with one point? If so, provide an input x_* such that knowing the point $(x_*, f(x_*))$ allows us to uniquely determine $f(x)$, and show how $f(x)$ can be determined from this point. If not, provide a proof of why this is not possible.

Solution. Yes, given the restrictions on the coefficients, it is possible for us to unique determine $f(x)$ using only one point.

To do this, we can pick $x_* = 100$ in order to do this. We observe here that since $0 \leq c_i < 100$, then we know that $x^{i+1} > 99x^i + 99x^{i-1} + \ldots + 99x^0$ for $x = 100$. With this in mind, in order to uniquely determine $f(x)$, we can follow the algorithm below:

1. Start by subtracting x^k from $f(100)$.
2. Then, from here, start subtracting x^{k-1} from the value.
3. The coefficient c_{k-1} is the number of times we subtract x^{k-1} from the value before it becomes negative.
4. Now, we repeat this process for $x^{k-2}, x^{k-3}, \ldots, x^0$. This will yield us the coefficients $c_{k-2}, c_{k-3}, \ldots, c_0$.

Thus, after this process, we will have uniquely determined $f(x)$ using only one point. ■

3 The CRT and Lagrange Interpolation

Let $n_1, \ldots n_k$ be pairwise co-prime, i.e. n_i and n_j are co-prime for all $i \neq j$. The Chinese Remainder Theorem (CRT) tells us that there exist solutions to the following system of congruences:

$$x \equiv a_1 \pmod{n_1} \tag{1}$$
$$x \equiv a_2 \pmod{n_2} \tag{2}$$
$$\vdots \tag{\vdots}$$
$$x \equiv a_k \pmod{n_k} \tag{k}$$

and all solutions are equivalent $\pmod{n_1 n_2 \cdots n_k}$. For this problem, parts (a)-(c) will walk us through a proof of the Chinese Remainder Theorem. We will then use the CRT to revisit Lagrange interpolation.

Problem 3.1. We start by proving the $k = 2$ case: Prove that we can always find an integer x_1 that solves (1) and (2) with $a_1 = 1, a_2 = 0$. Similarly, prove that we can always find an integer x_2 that solves (1) and (2) with $a_1 = 0, a_2 = 1$.

Solution. To begin with, we will find an integer x_1 which solves (1) and (2) with $a_1 = 1, a_2 = 0$. We observe that since $\gcd n_1, n_2 = 1$, then we know that n_2 has a multiplicative inverse such that $n_2(n_2^{-1}) \equiv 1 \pmod{n_1}$. With this in mind, let $x_1 = n_2(n_2^{-1})$. Then, we observe the following:

$$x_1 = n_2(n_2^{-1}) \equiv 1 \pmod{n_1}$$
$$x_1 = n_2(n_2^{-1}) \equiv 0 \pmod{n_2}$$

Similarly, we can find an integer x_2 such that $a_1 = 0$ and $a_2 = 1$. This can be done by letting x_2 be $n_1(n_1^{-1})$, yielding us the following:

$$x_2 = n_1(n_1^{-1}) \equiv 0 \pmod{n_1}$$
$$x_2 = n_1(n_1^{-1}) \equiv 1 \pmod{n_2}$$

■

Problem 3.2. Use part (a) to prove that we can always find at least one solution to (1) and (2) for any a_1, a_2. Furthermore, prove that all possible solutions are equivalent $\pmod{n_1 n_2}$.

Solution. Let us suppose that there exists some integer x such that:

$$x = a_1 n_2(n_2^{-1}) + a_2 n_1(n_1^{-1}) \pmod{n_1 n_2}$$

And we want to show that x also satisfies the following system of congruence:

$$x \equiv a_1 \pmod{n_1}$$
$$x \equiv a_2 \pmod{n_2}$$

Using part (a), we see that for any values of a_1, a_2, we have the following:

$$x_1 = a_1 n_2(n_2^{-1}) \equiv a_1 \pmod{n_1}$$
$$x_1 = a_2 n_2(n_2^{-1}) \equiv 0 \pmod{n_2}$$

$$x_2 = a_1 n_1(n_1^{-1}) \equiv 0 \pmod{n_1}$$
$$x_2 = a_2 n_1(n_1^{-1}) \equiv a_2 \pmod{n_2}$$

We see then that x satisfies the system of congruence, as seen below:

$$\begin{aligned} x = a_1 n_2 (n_2^{-1}) + a_2 n_1 (n_1^{-1}) \equiv a_1 \pmod{n_1} \\ x = a_1 n_2 (n_2^{-1}) + a_2 n_1 (n_1^{-1}) \equiv a_2 \pmod{n_2} \end{aligned}$$

Thus, we can conclude that ew can always find at least one solution to (1) and (2) for any a_1, a_2.

Now, we want to prove that x is a unique solution $\pmod{n_1 n_2}$. To do this, suppose that there is some integer y such that $y \equiv a_1 \pmod{n_1}$ and $y \equiv a_2 \pmod{n_2}$. This then implies that $y - x \equiv 0 \pmod{n_1}$ and $y - x \equiv 0 \pmod{n_2}$. Since $\gcd(n_1, n_2) = 1$, we can thus conclude then that $y - x \equiv 0 \pmod{n_1 n_2}$. In other words, $y \equiv x \pmod{n_1 n_2}$.

Therefore, we have shown that all possible solutions are equivalent $\pmod{n_1 n_2}$ as well.

■

Problem 3.3. Now we can tackle the case of arbitrary k: Use part (b) to prove that there exists a solution x to (1)-(k) and that this solution is unique $\pmod{n_1 n_2 \cdots n_k}$.

Solution. Using part (b), we know that the first two equations in the system

$$\begin{aligned} x &\equiv a_1 \pmod{n_1} && (1) \\ x &\equiv a_2 \pmod{n_2} && (2) \\ &\vdots && (\vdots) \\ x &\equiv a_k \pmod{n_k} && (k) \end{aligned}$$

has a solution of $x \equiv a_1 n_2 (n_2^{-1}) + a_2 n_1 (n_1^{-1}) \pmod{n_1 n_2}$. We see then that we can replace (1) and (2) with a single equation instead, reducing our system of congruences from k equations to $k-1$ equations.

Furthermore, note that since $n_1, \ldots, n_k$ are all pairwise coprime, it follows then that $n_1 n_2$ is also pairwise coprime to all other $n_3, \ldots, n_k$.

Then, from here, we can simply repeat this process of combining two equations into one, which eventually leads to the solution of the original system of congruence. ■

Problem 3.4. For polynomials $p_1(x)$, $p_2(x)$ and $q(x)$ we say that $p_1(x) \equiv p_2(x) \bmod q(x)$ if $p_1(x) - p_2(x)$ is of the form $q(x) \times m(x)$ for some polynomial $m(x)$.

Define the polynomials $x - a$ and $x - b$ to be co-prime if they have no common divisor of degree 1. Assuming that the CRT still holds when replacing x, a_i and n_i with polynomials (using the definition of co-prime polynomials just given), show that the system of congruences

$$\begin{aligned} p(x) &\equiv y_1 \pmod{(x - x_1)} && (1') \\ p(x) &\equiv y_2 \pmod{(x - x_2)} && (2') \\ &\vdots && (\vdots) \\ p(x) &\equiv y_k \pmod{(x - x_k)} && (k') \end{aligned}$$

has a unique solution $\pmod{(x - x_1) \cdots (x - x_k)}$ whenever the x_i are pairwise distinct. What is the connection to Lagrange interpolation?

Solution. We observe that since x_i are pairwise distinct, it follows then that $x-x_1, \ldots, x-x_i$ are all pairwise coprime. From here, we can utilise the Chinese Remainder to state that there exists some polynomial $p(x)$ which satisfies all equations $(1'), \ldots, (k')$ and has a unique solution $\pmod{(x-x_1)\cdots(x-x_i)}$.

Now, to actually find this solution, see that with CRT, we know that for $N = \prod_{i=1}^{k} n_i$, we have an x which satisfies the congruences:

$$\begin{aligned} x &\equiv a_1 \pmod{n_1} \\ &\vdots \\ x &\equiv a_i \pmod{n_1} \\ &\vdots \\ x &\equiv a_k \pmod{n_k} \end{aligned}$$

Furthermore, we have that $x \equiv \sum_{i=1}^{k} a_i b_i \pmod{N}$, where $b_i = \frac{N}{n_i}\left(\frac{N}{n_i}\right)^{-1}$. From this, we see that $b \equiv 1 \pmod{n_i}$ for $b \equiv 0 \pmod{n_0}$ for $i \neq j$.

Now, a similar idea can be extended to polynomials: suppose there's a polynomial $p_i(x)$ such that for $i \neq j$, we have

$$\begin{aligned} p_i(x_i) &= 1 \\ p_i(x_j) &= 0 \end{aligned}$$

Then, we observe that the polynomial that satisfies this would be:

$$p_i = \frac{(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_k)}{(x_i-x_1)\cdots(x_i-x_k)}$$

Then, from this, we observe that for x_i, we get that $p_i = 1$. But, for any x_j where $j \neq i$, we see that $p_i = 0$.

Now, from here, we see that $p(x) = y_1 p_1(x) + \ldots + y_k p_k(x)$. However, observe that this is just simply Lagrange Interpolation; thus, we see that not only is there a polynomial that satisfies the system of congruences given, but that there is also a connection between CRT and Lagrange Interpolation. ■

4 Trust No One

Problem 4.1. Gandalf has assembled a fellowship of nine peoples to transport the One Ring to the fires of Mount Doom: five humans, two hobbits, one elf, and one dwarf. The ring has great power that may be of use to the fellowship during their long and dangerous journey. Unfortunately, the use of its immense power will eventually corrupt the user, so it must not be used except in the most dire of circumstances. To safeguard against this possibility, Gandalf wishes to keep the instructions a secret from members of the fellowship. The secret must only be revealed if enough members of the fellowship are present and agree to use it.

Gandalf has hired your services to help him come up with a secret sharing scheme that accomplishes this task, summarized by the following points:

- There is a party of five humans, two hobbits, an elf, and a dwarf, and a secret message that must remain unknown to everyone if not enough members of the party agree.
- A group of people consisting of at least two people from different people classes and at least one people class that is fully represented (i.e., has all members present) can unlock the secret of the ring.

A few examples: only five humans agreeing to use the ring is not enough to know the instructions. One hobbit and four humans is not enough. However, all five humans and one hobbit agreeing is enough. Both hobbits and the dwarf agreeing is enough.

Solution. Let s denote the secret message which Gandalf wants to share. Then, let us split the message into four shares: $s_1, \ldots, s_4$.

Furthermore, we make it so that any 2 shares of the message is enough to reconstruct the original secret message s.

Now, from here, we can split the people into groups G_i based on their classes:

- G_1 is humans,
- G_2 is hobbits,
- G_3 is elves, and
- G_4 is dwarfs.

From here, what we can do is to split each share s_i of the message into n parts, where n is the size of G_i.

So, the message is now currently split into nine parts, and let us denote these parts as t_i. We will then distribute these shares as follow:

- Let $t_1, \ldots, t_5$ be given to the humans,
- t_6 and t_7 be given to the hobbits,
- t_8 to the elf, and
- t_9 to the dwarf.

Now, each member of each group will hold onto a unique t_i. In other words, to reconstruct, for example, s_1, we need all humans present (since human 1 holds t_1, human 2 holds t_2, etc.).

However, if we stop here, we observe that in order to reconstruct s, we need at least two groups that are fully represented; Gandalf requires only one group.

So, from here, we will also start giving each group shares that are from other groups. However, we will leave out one share from each group. So, for example, the elf will have $t_1, \ldots, t_4$ and t_6, alongside t_8 (the elf doesn't have t_9 because there is only one share for the dwarf). Meanwhile, say Human 1 will have only shares t_1 and t_6.

This will ensure that the only way to get a share s_i of a message is to have all members of at least one group G_i be present. Furthermore, as we need two shares to reconstruct s, it means that we have to have two different people classes. ■

5 Secret Sharing With Spies

Problem 5.1. An officer stored an important letter in her safe. In case she becomes unreachable in battle, she decides to share the password (which is a number) with her troops. However, everyone knows that there are 3 spies among the troops, but no one knows who they are except for the three spies themselves. The 3 spies can coordinate with each other and they will either lie and make people not able to open the safe, or will open the safe themselves if they can. Therefore, the officer would like a scheme to share the password that satisfies the following conditions:

- When N of them get together, they are guaranteed to be able to open the safe even if they have spies among them.
- The 3 spies must not be able to open the safe all by themselves.

Please help the officer to design a scheme to share her password. What is the scheme? What is the smallest N? Show your work and argue why your scheme works and any smaller N couldn't work. (The troops only have one chance to open the safe; if they fail the safe will self-destruct.)

Solution. Let s denote the secret message. Furthermore, suppose we are working in $\mathrm{GF}(p)$, for some prime p where $p > s$ and $p > N$. The officer can then distribute $n = k$ shares to the k troops (i.e. they give $P(k)$ to the kth troop).

Now, let us construct a polynomial such that $P(0) = s$. Next, in order to ensure that the three spies can't open the safe by themselves, we must have a polynomial of degree at least 3. Then, our polynomial is in the form $P(x) = a_3x^3 + a_2x^2 + a_1x + s$.

Since $P(x)$ is of degree 3, then in order to figure out the secret, we need at least 4 people to create the unique polynomial.

Now, we want to find some minimum N which guarantees that the people are able to open the safe even if there are spies among them. This is essentially a general error situation, where the spies can be viewed as the points where the message is being corrupted. With this in mind, we see then that to ensure the officer's first condition, we need $N = 4 + 2(3) = 10$ people.

We know that with $N = 10$ people, it is possible to reconstruct $P(x)$. We observe that for any given subset of 7 values of i between 1 and 10, there is a unique polynomial $P(x)$ such that $P(i) = r_i$ at these values of i.

To see this, suppose that $P'(x)$ is any other polynomial with degree 3 that goes through these 7 points. Then among these 7 points, there are at most 3 errors, and therefore on at least n of the points we must have that $P'(i) = P(i)$. Since this is the case, it then follows that $P'(x)$ and $P(x)$ must be the same polynomial, as a polynomial of degree 3 is defined uniquely by 4 points.

To see that our scheme works, we observe that finding the message is the same as the $N = 11$ people coming together to reconstruct $P(x)$ using 11 points. The issue then becomes how to find the error points; i.e., the values which the spies are lying about.

In order to find the error points, the troops can simply use the Berlekamp-Welch Algorithm in order to reconstruct the original polynomial $P(x)$. From there, they can find $P(0)$, allowing them to find the secret message of the officer.

Now, to see why the minimum number is $N = 11$, we observe here that if we had $N < 11$ people, then using the Berlekamp-Welch Algorithm, we'd end with with less equations than there are unknown variables; in this case, it'll result in us being unable to solve for the system of equations, and thus be unable to reconstruct $P(x)$. ■

6 Error-Correcting Codes

Problem 6.1. Recall from class the error-correcting code for erasure errors, which protects against up to k lost packets by sending a total of $n + k$ packets (where n is the number of packets in the original message). Often the number of packets lost is not some fixed number k, but rather a *fraction* of the number of packets sent. Suppose we wish to protect against a fraction α of lost packets (where $0 < \alpha < 1$). At least how many packets do we need to send (as a function of n and α)?

Solution. In order to protect against a fraction α of lost packets $0 < \alpha < 1$, we observe that we need to find an x such that $\alpha(n + x) = n$:

$$\begin{aligned}\alpha(n + x) &= n\\ n + x &= \frac{n}{\alpha}\\ x &= \frac{n}{\alpha} - n\\ &= n(\frac{1}{\alpha} - 1)\end{aligned}$$

Thus, by sending an additional $n(\frac{1}{\alpha} - 1)$ packets, we can protect against the loss; i.e. we must send $n + n(\frac{1}{\alpha} - 1) = \frac{n}{\alpha}$ packets. ■

Problem 6.2. Repeat part (a) for the case of general errors.

Solution. In the case of general errors, we will need to find some x such that $\alpha(n + x) = n + 2\alpha n$:

$$\begin{aligned}\alpha(n + x) &= n + 2\alpha n\\ n + x &= \frac{n}{\alpha} + 2n\\ x &= \frac{n}{\alpha} + n\end{aligned}$$

Thus, in the case of general errors, we have to send an additional $\frac{n}{\alpha} + n$ packets in order to safeguard against them. In other words, we have to send a total of $n + \frac{n}{\alpha} + n$ packets. ■

7 Alice and Bob

Problem 7.1. Alice decides that instead of encoding her message as the values of a polynomial, she will encode her message as the coefficients of a degree 2 polynomial $P(x)$. For her message $[m_1, m_2, m_3]$, she creates the polynomial $P(x) = m_1x^2 + m_2x + m_3$ and sends the five packets $(0, P(0))$, $(1, P(1))$, $(2, P(2))$, $(3, P(3))$, and $(4, P(4))$ to Bob. However, one of the packet y-values is changed by Eve before it reaches Bob. If Bob receives

$$(0,1),(1,3),(2,0),(3,1),(4,0)$$

and knows Alice's encoding scheme and that Eve changed one of the packets, can he recover the original message? If so, find it as well as the x-value of the packet that Eve changed. If he can't, explain why. Work in mod 7.

Solution. To figure out the original message, let $Q(x) = a_3x^3 + a_2x^2 + a_1x + a_0$. Now, let us create the following system:

$$\begin{aligned} a_0 - b_0 &= 0 \\ a_3 + a_2 + a_1 + a_0 - 3b_0 &= 3 \\ 8a_3 + 4a_2 + 2a_1 + a_0 &= 0 \\ 27a_3 + 9a_2 + 3a_1 + a_0 - b_0 &= 3 \\ 64a_3 + 16a_2 + 4a_1 + a_0 &= 0 \end{aligned}$$

■

Problem 7.2. Bob gets tired of decoding degree 2 polynomials. He convinces Alice to encode her messages on a degree 1 polynomial. Alice, just to be safe, continues to send 5 points on her polynomial even though it is only degree 1. She makes sure to choose her message so that it can be encoded on a degree 1 polynomial. However, Eve changes two of the packets. Bob receives $(0, 5)$, $(1, 7)$, $(2, x)$, $(3, 5)$, $(4, 0)$. If Alice sent $(0, 5)$, $(1, 7)$, $(2, 9)$, $(3, -2)$, $(4, 0)$, for what values of x will Bob not uniquely be able to determine Alice's message? Assume that Bob knows Eve changed two packets. Work in mod 13.

Solution. We observe that $-2 \equiv 11 \pmod{13}$. Then, the polynomial which Alice used is $P(x) = 2x + 5$.

Now, in order to determine the value x in which Bob will not be able to uniquely determine Alice's message, we have to find some polynomial such that it connects three of the points which Bob receives linearly, and also includes both of the error point.

Then, we must find some polynomial that connects $(1, 7), (2, x), (3, 5)$ or $(2, x), (3, 5), (4, 0)$ which isn't the polynomial Alice has.

In this case, we observe that $x = 6$ and $x = 10$ satisfies this; in the case of $x = 6$, we see that we can have a polynomial $P'(x) = -x + 8$. In the case of $x = 10$, we have the polynomial $P'(x) = -5x + 20$.

Remark 7.1. For my solution, I assumed that a polynomial of degree 0 wouldn't work. In other words, if $x = 5$, then Bob could end up with a polynomial $P'(x) = 5$. However, since Alice is encoding messages on a degree 1 polynomial, then Bob knows that $P'(x) = 5$ would not be correct; thus, he can still deduce the correct polynomial $P(x) = 2x + 5$.

■

Problem 7.3. Alice wants to send a length 9 message to Bob. There are two communication channels available to her: Channel X and Channel Y. When n packets are fed through Channel X, only 6 packets, picked arbitrarily, are delivered. Similarly, Channel Y will only deliver 6 packets, picked arbitrarily, but it will also corrupt (change the value) of one of the delivered packets. Each channel will only work if at least 10 packets are sent through it. Using each of the two channels once, provide a way for Alice to send her message to Bob so that he can always reconstruct it.

Solution. In order to do this, let us construct a polynomial $p(x)$ of degree 8, where the first 9 points $p(0), \ldots, p(8)$ contains the message Alice wants to send.

Now, Alice must send the first 10 points of $p(x)$ through Channel X, then the next 10 points through Channel Y. Then, through Channel X, only 6 points will get through, and for Channel Y, six different points (with one that's corrupted) makes it through.

Since we chose distinct points, this ensures that we won't have a situation where the six points which make it through each Channel ends up being the same. Furthermore, since Bob now has 12 points, this will then ensure that Bob will be able to always reconstruct the message which Alice wants to send. ■