Today: Error Correction.

...from Polynomials.

Give out \( n \) points on degree \( k - 1 \) polynomial \( P(x) \).

Encode secret as \( y \)-intercept of \( P(x) \).

Any \( k \) can reconstruct polynomial.

\((d+1)\)-points correspond to exactly one Polynomial.

Lagrange Interpolation \( \Rightarrow \) there exists a polynomial that hits points.

Polynomial Division \( \Rightarrow \) there is only one.

Can also reconstruct polynomial using a linear system.
Today: Error Correction.

...from Polynomials.
Secret Sharing.
Today: Error Correction.

...from Polynomials.

Secret Sharing.
  Give out \( n \) points on degree \( k - 1 \) polynomial \( P(x) \)
...from Polynomials.

Secret Sharing.

Give out $n$ points on degree $k - 1$ polynomial $P(x)$
Encode secret as $y$-intercept of $P(x)$. 
Secret Sharing.
Give out $n$ points on degree $k - 1$ polynomial $P(x)$
  Encode secret as $y$-intercept of $P(x)$.
Any $k$ can reconstruct polynomial.
Today: Error Correction.

...from Polynomials.

Secret Sharing.
  Give out \( n \) points on degree \( k - 1 \) polynomial \( P(x) \)
  Encode secret as \( y \)-intercept of \( P(x) \).
  Any \( k \) can reconstruct polynomial.
Today: Error Correction.

...from Polynomials.

Secret Sharing.
  Give out $n$ points on degree $k - 1$ polynomial $P(x)$.
  Encode secret as $y$-intercept of $P(x)$.
  Any $k$ can reconstruct polynomial.

$(d + 1)$-points correspond to exactly one Polynomial.
...from Polynomials.

Secret Sharing.
- Give out \( n \) points on degree \( k - 1 \) polynomial \( P(x) \)
- Encode secret as \( y \)-intercept of \( P(x) \).
- Any \( k \) can reconstruct polynomial.

\((d + 1)\)-points correspond to exactly one Polynomial.

Lagrange Interpolation \( \implies \) there exists a polynomial that hits points.
Today: Error Correction.

...from Polynomials.

Secret Sharing.
  Give out \( n \) points on degree \( k - 1 \) polynomial \( P(x) \).
  Encode secret as \( y \)-intercept of \( P(x) \).
  Any \( k \) can reconstruct polynomial.

\((d + 1)\)-points correspond to exactly one Polynomial.

Lagrange Interpolation \( \implies \) there exists a polynomial that hits points.
Polynomial Division \( \implies \) there is only one.
...from Polynomials.

Secret Sharing.
  Give out $n$ points on degree $k - 1$ polynomial $P(x)$
  Encode secret as $y$-intercept of $P(x)$.
  Any $k$ can reconstruct polynomial.

$(d + 1)$-points correspond to exactly one Polynomial.

Lagrange Interpolation $\implies$ there exists a polynomial that hits points.
Polynomial Division $\implies$ there is only one.
  Can also reconstruct polynomial using a linear system.
Erasure Codes.

Satellite

GPS device
Erasure Codes.

Satellite sends 6 packets.
3 packet message.

GPS device gets packets 1, 2, and 3.
Erasure Codes.

Satellite

3 packet message.

GPS device

Lose 3 out 6 packets.
Erasure Codes.

Satellite

3 packet message. So send 6!

Lose 3 out 6 packets.

GPS device
Erasure Codes.

3 packet message. So send 6!

Lose 3 out 6 packets.
Erasure Codes.

Satellite

3 packet message. So send 6!

Lose 3 out 6 packets.
Erasure Codes.

Satellite

3 packet message. So send 6!

Lose 3 out 6 packets.

Gets packets 1, 1, and 3.
Solution Idea.

\[ n \] packet message, channel that loses \( k \) packets.
Solution Idea.

\( n \) packet message, channel that loses \( k \) packets.
Must send \( n + k \) packets!
Solution Idea.

\( n \) packet message, channel that loses \( k \) packets.

Must send \( n + k \) packets!

- Any \( n \) packets
Solution Idea.

$n$ packet message, channel that loses $k$ packets.
Must send $n + k$ packets!

Any $n$ packets
should allow reconstruction of $n$ packet message.
Solution Idea.

An $n$ packet message, channel that loses $k$ packets.
Must send $n + k$ packets!

Any $n$ packets should allow reconstruction of $n$ packet message.

Any $n$ point values
Solution Idea.

$n$ packet message, channel that loses $k$ packets.

Must send $n + k$ packets!

Any $n$ packets should allow reconstruction of $n$ packet message.

Any $n$ point values allow reconstruction of degree $n - 1$ polynomial.
Solution Idea.

\( n \) packet message, channel that loses \( k \) packets.
Must send \( n + k \) packets!

Any \( n \) packets
should allow reconstruction of \( n \) packet message.

Any \( n \) point values
allow reconstruction of degree \( n - 1 \) polynomial.

Alright!
Solution Idea.

$n$ packet message, channel that loses $k$ packets.
Must send $n + k$ packets!

Any $n$ packets should allow reconstruction of $n$ packet message.

Any $n$ point values allow reconstruction of degree $n - 1$ polynomial.

Alright!!
Solution Idea.

\( n \) packet message, channel that loses \( k \) packets.
Must send \( n + k \) packets!

Any \( n \) packets
should allow reconstruction of \( n \) packet message.

Any \( n \) point values
allow reconstruction of degree \( n - 1 \) polynomial.

Alright!!!
Solution Idea.

Any $n$ packets should allow reconstruction of a $n$-packet message.

Any $n$ point values allow reconstruction of a degree $n-1$ polynomial.

Alright!!!!
Solution Idea.

\[ n \] packet message, channel that loses \( k \) packets.
Must send \( n + k \) packets!

Any \( n \) packets
should allow reconstruction of \( n \) packet message.

Any \( n \) point values
allow reconstruction of degree \( n - 1 \) polynomial.

Alright!!!!!
Solution Idea.

\( n \) packet message, channel that loses \( k \) packets.

Must send \( n + k \) packets!

Any \( n \) packets
should allow reconstruction of \( n \) packet message.

Any \( n \) point values
allow reconstruction of degree \( n - 1 \) polynomial.

Alright!!!!!!
Solution Idea.

$n$ packet message, channel that loses $k$ packets.

Must send $n + k$ packets!

Any $n$ packets should allow reconstruction of $n$ packet message.

Any $n$ point values allow reconstruction of degree $n - 1$ polynomial.

Alright!!!!!!
Solution Idea.

$n$ packet message, channel that loses $k$ packets.
Must send $n + k$ packets!

Any $n$ packets should allow reconstruction of $n$ packet message.

Any $n$ point values allow reconstruction of degree $n - 1$ polynomial.

Alright!!!!!!

Use polynomials.
Problem: Want to send a message with $n$ packets.
Problem: Want to send a message with $n$ packets.
Channel: Lossy channel: loses $k$ packets.
Erasure Code.

**Problem:** Want to send a message with $n$ packets.

**Channel:** Lossy channel: loses $k$ packets.

**Question:** Can you send $n + k$ packets and recover message?
Problem: Want to send a message with $n$ packets.
Channel: Lossy channel: loses $k$ packets.
Question: Can you send $n + k$ packets and recover message?
A degree $n - 1$ polynomial determined by any $n$ points!
Problem: Want to send a message with \( n \) packets.

Channel: Lossy channel: loses \( k \) packets.

Question: Can you send \( n + k \) packets and recover message?

A degree \( n - 1 \) polynomial determined by any \( n \) points!

Erasure Coding Scheme: message = \( m_0, m_2, \ldots, m_{n-1} \).

1. Choose prime \( p \approx 2^b \) for packet size \( b \).

2. \( P(x) = m_{n-1}x^{n-1} + \cdots m_0 \pmod{p} \).

3. Send \( P(1), \ldots, P(n+k) \).
Problem: Want to send a message with \( n \) packets.

Channel: Lossy channel: loses \( k \) packets.

Question: Can you send \( n+k \) packets and recover message?

A degree \( n-1 \) polynomial determined by any \( n \) points!

Erasure Coding Scheme: message = \( m_0, m_2, \ldots, m_{n-1} \).

1. Choose prime \( p \approx 2^b \) for packet size \( b \).
2. \( P(x) = m_{n-1}x^{n-1} + \cdots m_0 \pmod{p} \).
3. Send \( P(1), \ldots, P(n+k) \).

Any \( n \) of the \( n+k \) packets gives polynomial ...
Problem: Want to send a message with \( n \) packets.

Channel: Lossy channel: loses \( k \) packets.

Question: Can you send \( n + k \) packets and recover message?

A degree \( n - 1 \) polynomial determined by any \( n \) points!

Erasure Coding Scheme: message = \( m_0, m_2, \ldots, m_{n-1} \).

1. Choose prime \( p \approx 2^b \) for packet size \( b \).
2. \( P(x) = m_{n-1}x^{n-1} + \cdots m_0 \pmod{p} \).
3. Send \( P(1), \ldots, P(n+k) \).

Any \( n \) of the \( n+k \) packets gives polynomial ...and message!
Erasure Codes.

Satellite

GPS device
Erasure Codes.

Satellite

GPS device

n packet message.

Any n packets is enough!
Erasure Codes.

Satellite $n$ packet message.

Lose $k$ packets.

GPS device
Erasure Codes.

Satellite

1 2 \ldots n+k

GPS device

$n$ packet message. So send $n+k$!

Lose $k$ packets.
Erasure Codes.

Satellite

1 2 ⋯ n+k

1 2 ⋯ n+k

GPS device

$n$ packet message. So send $n+k$!

Lose $k$ packets.
Erasure Codes.

Satellite

1 2 ······· n+k

GPS device

n packet message. So send $n+k$!

Lose $k$ packets.
Erasure Codes.

Satellite

1 2 ⋯ $n+k$

GPS device

$n$ packet message. So send $n+k$!

Lose $k$ packets.

Any $n$ packets is enough!
Erasure Codes.

Satellite

\[ 1 \quad 2 \quad \cdots \quad n+k \]

GPS device

\[ 1 \quad 2 \quad \cdots \quad n+k \]

*n packet message. So send \( n+k \)!

Lose \( k \) packets.

Any \( n \) packets is enough!
Erasure Codes.

Satellite

\[
1 \quad 2 \quad \cdots \quad n+k
\]

GPS device

\[
1 \quad 2 \quad \cdots \quad n+k
\]

\[\text{Lose } k \text{ packets.}\]

Any \( n \) packets is enough!

\[\text{Optimal.}\]
Size: Can choose a prime between $2^{b-1}$ and $2^b$. (Lose at most 1 bit per packet.)
Size: Can choose a prime between $2^{b-1}$ and $2^b$. (Lose at most 1 bit per packet.)

But: packets need label for $x$ value.
Size: Can choose a prime between $2^{b-1}$ and $2^b$. (Lose at most 1 bit per packet.)

But: packets need label for $x$ value.

There are Galois Fields $GF(2^n)$ where one loses nothing.
Information Theory.

Size: Can choose a prime between $2^{b-1}$ and $2^b$. (Lose at most 1 bit per packet.)

But: packets need label for $x$ value.

There are Galois Fields $GF(2^n)$ where one loses nothing.

– Can also run the Fast Fourier Transform.
Information Theory.

Size: Can choose a prime between $2^{b-1}$ and $2^b$. (Lose at most 1 bit per packet.)
But: packets need label for $x$ value.
There are Galois Fields $GF(2^n)$ where one loses nothing.
– Can also run the Fast Fourier Transform.
In practice, $O(n)$ operations with almost the same redundancy.
Size: Can choose a prime between $2^{b-1}$ and $2^b$. (Lose at most 1 bit per packet.)

But: packets need label for $x$ value.

There are Galois Fields $GF(2^n)$ where one loses nothing.

– Can also run the Fast Fourier Transform.

In practice, $O(n)$ operations with almost the same redundancy.

Comparison with Secret Sharing: information content.
Size: Can choose a prime between $2^{b-1}$ and $2^b$. (Lose at most 1 bit per packet.)
But: packets need label for $x$ value.
There are Galois Fields $GF(2^n)$ where one loses nothing.
– Can also run the Fast Fourier Transform.
In practice, $O(n)$ operations with almost the same redundancy.
Comparison with Secret Sharing: information content.
  Secret Sharing: each share is size of whole secret.
Information Theory.

Size: Can choose a prime between $2^{b-1}$ and $2^b$. (Lose at most 1 bit per packet.)

But: packets need label for $x$ value.

There are Galois Fields $GF(2^n)$ where one loses nothing.

– Can also run the Fast Fourier Transform.

In practice, $O(n)$ operations with almost the same redundancy.

Comparison with Secret Sharing: information content.

  Secret Sharing: each share is size of whole secret.
  Coding: Each packet has size $1/n$ of the whole message.
Size: Can choose a prime between $2^{b-1}$ and $2^b$. (Lose at most 1 bit per packet.)

But: packets need label for $x$ value.

There are Galois Fields $GF(2^n)$ where one loses nothing.

– Can also run the Fast Fourier Transform.

In practice, $O(n)$ operations with almost the same redundancy.

Comparison with Secret Sharing: information content.

  Secret Sharing: each share is size of whole secret.
  Coding: Each packet has size $1/n$ of the whole message.
Erasure Code: Example.

Send message of 1, 4, and 4.

How?
Lagrange Interpolation.
Linear System.
Work modulo 5.

\[ P(x) = x^2 \pmod{5} \]

\[ P(1) = 1, \ P(2) = 4, \ P(3) = 9 = 4 \pmod{5} \]

Send \((0, P(0)) \ldots (5, P(5))\).

Why?
Better work modulo 7 at least!
Erasure Code: Example.

Send message of 1, 4, and 4.
Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$. 
Erasure Code: Example.

Send message of 1, 4, and 4.
Make polynomial with \( P(1) = 1, P(2) = 4, P(3) = 4 \). How?
Erasure Code: Example.

Send message of 1, 4, and 4.
Make polynomial with \( P(1) = 1, \ P(2) = 4, \ P(3) = 4 \).
How?
   Lagrange Interpolation.
Send message of 1, 4, and 4.

Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$.

How?

Lagrange Interpolation.
Linear System.
Erasure Code: Example.

Send message of 1, 4, and 4.

Make polynomial with \( P(1) = 1, \ P(2) = 4, \ P(3) = 4 \).

How?

   Lagrange Interpolation.
   Linear System.
Erasure Code: Example.

Send message of 1, 4, and 4.
Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$.

How?
- Lagrange Interpolation.
- Linear System.
- Work modulo 5.

Better work modulo 7 at least!

Why?

Send $(0, P(0))$ . . . $(5, P(5))$. 6 points.
Erasure Code: Example.

Send message of 1, 4, and 4.
Make polynomial with $P(1) = 1, P(2) = 4, P(3) = 4$.
How?
Lagrange Interpolation.
Linear System.
Work modulo 5.
$P(x) = x^2 \pmod{5}$
Send message of 1, 4, and 4.

Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$.

How?

Lagrange Interpolation.
Linear System.

Work modulo 5.

$P(x) = x^2 \pmod{5}$

$P(1) = 1$, 

Better work modulo 7 at least!
Send message of 1, 4, and 4.

Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$.

How?

Lagrange Interpolation.
Linear System.

Work modulo 5.

$P(x) = x^2 \pmod{5}$

$P(1) = 1$, $P(2) = 4$, $P(3) = 4$. 
Erasure Code: Example.

Send message of 1, 4, and 4.
Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$.

How?

Lagrange Interpolation.
Linear System.

Work modulo 5.

$P(x) = x^2 \pmod{5}$

$P(1) = 1, P(2) = 4, P(3) = 9 = 4 \pmod{5}$
Erasure Code: Example.

Send message of 1, 4, and 4.

Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$.

How?

Lagrange Interpolation.
Linear System.

Work modulo 5.

$P(x) = x^2 \pmod{5}$

$P(1) = 1$, $P(2) = 4$, $P(3) = 9 = 4 \pmod{5}$
Erasure Code: Example.

Send message of 1, 4, and 4.
Make polynomial with \( P(1) = 1, P(2) = 4, P(3) = 4 \).
How?
Lagrange Interpolation.
Linear System.
Work modulo 5.

\[
P(x) = x^2 \pmod 5
\]

\[
P(1) = 1, P(2) = 4, P(3) = 9 = 4 \pmod 5
\]

Send \((0, P(0)) \ldots (5, P(5))\).
Erasure Code: Example.

Send message of 1, 4, and 4.
Make polynomial with \( P(1) = 1, \ P(2) = 4, \ P(3) = 4 \).
How?

Lagrange Interpolation.
Linear System.

Work modulo 5.

\[
P(x) = x^2 \pmod{5}
\]

\[
P(1) = 1, \ P(2) = 4, \ P(3) = 9 = 4 \pmod{5}
\]

Send \((0, P(0))\ldots(5, P(5))\).
Send message of 1, 4, and 4.

Make polynomial with \( P(1) = 1 \), \( P(2) = 4 \), \( P(3) = 4 \).

How?

Lagrange Interpolation.
Linear System.

Work modulo 5.

\[
P(x) = x^2 \pmod{5}
\]

\[
P(1) = 1, P(2) = 4, P(3) = 9 = 4 \pmod{5}
\]

Send \((0, P(0)) \ldots (5, P(5))\).

6 points.
Erasure Code: Example.

Send message of 1, 4, and 4.
Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$.
How?

Lagrange Interpolation.
Linear System.

Work modulo 5.

$P(x) = x^2 \pmod{5}$

$P(1) = 1$, $P(2) = 4$, $P(3) = 9 = 4 \pmod{5}$

Send $(0, P(0)) \ldots (5, P(5))$.

6 points. Better work modulo 7 at least!
Erasure Code: Example.

Send message of 1, 4, and 4.
Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$.
How?

Lagrange Interpolation.
Linear System.

Work modulo 5.

\[ P(x) = x^2 \pmod{5} \]

$P(1) = 1$, $P(2) = 4$, $P(3) = 9 = 4 \pmod{5}$

Send $(0, P(0)) \ldots (5, P(5))$.

6 points. Better work modulo 7 at least!

Why?
Send message of 1, 4, and 4.

Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$.

How?

- Lagrange Interpolation.
- Linear System.

Work modulo 5.

$P(x) = x^2 \pmod{5}$

$P(1) = 1$, $P(2) = 4$, $P(3) = 9 = 4 \pmod{5}$

Send $(0, P(0)) \ldots (5, P(5))$.

6 points. Better work modulo 7 at least!

Why? $(0, P(0)) = (5, P(5)) \pmod{5}$
Example

Make polynomial with \( P(1) = 1, \ P(2) = 4, \ P(3) = 4. \)
Example

Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$.
Modulo 7 to accommodate at least 6 packets.
Example

Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$.
Modulo 7 to accommodate at least 6 packets.
Linear equations:

$P(x) = 2x^2 + 4x + 2$

Send Packets: $(1, 1), (2, 4), (3, 4), (4, 7), (5, 2), (6, 0)$

Notice that packets contain "x-values".
Example

Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$. Modulo 7 to accommodate at least 6 packets.
Linear equations:

$$P(1) = a_2 + a_1 + a_0 \equiv 1 \pmod{7}$$
Example

Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$.
Modulo 7 to accommodate at least 6 packets.

Linear equations:

\[
P(1) = a_2 + a_1 + a_0 \equiv 1 \pmod{7}
\]
\[
P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7}
\]
Example

Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$.
Modulo 7 to accommodate at least 6 packets.

Linear equations:

\[
P(1) = a_2 + a_1 + a_0 \equiv 1 \pmod{7}
\]
\[
P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7}
\]
\[
P(3) = 2a_2 + 3a_1 + a_0 \equiv 4 \pmod{7}
\]

Send packets: $(1, 1)$, $(2, 4)$, $(3, 4)$, $(4, 7)$, $(5, 2)$, $(6, 0)$.

Notice that packets contain "x-values."
Example

Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$.
Modulo 7 to accommodate at least 6 packets.
Linear equations:

\[
\begin{align*}
P(1) &= a_2 + a_1 + a_0 \equiv 1 \pmod{7} \\
P(2) &= 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \\
P(3) &= 2a_2 + 3a_1 + a_0 \equiv 4 \pmod{7}
\end{align*}
\]
Example

Make polynomial with \( P(1) = 1, \ P(2) = 4, \ P(3) = 4 \).
Modulo 7 to accommodate at least 6 packets.
Linear equations:

\[
\begin{align*}
P(1) &= a_2 + a_1 + a_0 \equiv 1 \pmod{7} \\
P(2) &= 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \\
P(3) &= 2a_2 + 3a_1 + a_0 \equiv 4 \pmod{7}
\end{align*}
\]

\[6a_1 + 3a_0 = 2 \pmod{7},\]
Example

Make polynomial with \( P(1) = 1, \ P(2) = 4, \ P(3) = 4. \)
Modulo 7 to accommodate at least 6 packets.
Linear equations:

\[
\begin{align*}
P(1) &= a_2 + a_1 + a_0 \equiv 1 \pmod{7} \\
P(2) &= 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \\
P(3) &= 2a_2 + 3a_1 + a_0 \equiv 4 \pmod{7}
\end{align*}
\]

\[
6a_1 + 3a_0 = 2 \pmod{7}, \quad 5a_1 + 4a_0 = 0 \pmod{7}
\]
Example

Make polynomial with \( P(1) = 1, P(2) = 4, P(3) = 4 \).
Modulo 7 to accommodate at least 6 packets.
Linear equations:

\[
\begin{align*}
P(1) &= a_2 + a_1 + a_0 \equiv 1 \pmod{7} \\
P(2) &= 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \\
P(3) &= 2a_2 + 3a_1 + a_0 \equiv 4 \pmod{7}
\end{align*}
\]

\[6a_1 + 3a_0 = 2 \pmod{7}, \quad 5a_1 + 4a_0 = 0 \pmod{7}\]
\[a_1 = 2a_0.\]
Example

Make polynomial with \( P(1) = 1, P(2) = 4, P(3) = 4 \).
Modulo 7 to accommodate at least 6 packets.
Linear equations:

\[
\begin{align*}
P(1) &= a_2 + a_1 + a_0 \equiv 1 \pmod{7} \\
P(2) &= 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \\
P(3) &= 2a_2 + 3a_1 + a_0 \equiv 4 \pmod{7}
\end{align*}
\]

\[
6a_1 + 3a_0 = 2 \pmod{7}, \quad 5a_1 + 4a_0 = 0 \pmod{7}
\]

\[
a_1 = 2a_0, \quad a_0 = 2 \pmod{7}
\]
Example

Make polynomial with \( P(1) = 1, P(2) = 4, P(3) = 4 \).
Modulo 7 to accommodate at least 6 packets.
Linear equations:

\[
\begin{align*}
P(1) &= a_2 + a_1 + a_0 \equiv 1 \pmod{7} \\
P(2) &= 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \\
P(3) &= 2a_2 + 3a_1 + a_0 \equiv 4 \pmod{7}
\end{align*}
\]

\[
\begin{align*}
6a_1 + 3a_0 &= 2 \pmod{7}, & 5a_1 + 4a_0 &= 0 \pmod{7} \\
a_1 &= 2a_0, & a_0 &= 2 \pmod{7} & a_1 &= 4 \pmod{7}
\end{align*}
\]
Example

Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$.
Modulo 7 to accommodate at least 6 packets.
Linear equations:

\[
P(1) = a_2 + a_1 + a_0 \equiv 1 \pmod{7}
\]
\[
P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7}
\]
\[
P(3) = 2a_2 + 3a_1 + a_0 \equiv 4 \pmod{7}
\]

\[
6a_1 + 3a_0 \equiv 2 \pmod{7}, \quad 5a_1 + 4a_0 \equiv 0 \pmod{7}
\]

\[
a_1 = 2a_0. \quad a_0 = 2 \pmod{7} \quad a_1 = 4 \pmod{7} \quad a_2 = 2 \pmod{7}
\]
Example

Make polynomial with \( P(1) = 1, \ P(2) = 4, \ P(3) = 4. \)
Modulo 7 to accommodate at least 6 packets.
Linear equations:

\[
\begin{align*}
P(1) &= a_2 + a_1 + a_0 \equiv 1 \pmod{7} \\
P(2) &= 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \\
P(3) &= 2a_2 + 3a_1 + a_0 \equiv 4 \pmod{7}
\end{align*}
\]

\[
\begin{align*}
6a_1 + 3a_0 &= 2 \pmod{7}, &
5a_1 + 4a_0 &= 0 \pmod{7} \\
a_1 &= 2a_0, &
a_0 &= 2 \pmod{7}, & a_1 &= 4 \pmod{7}, & a_2 &= 2 \pmod{7}
\end{align*}
\]

\[P(x) = 2x^2 + 4x + 2\]
Example

Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$. Modulo 7 to accommodate at least 6 packets.

Linear equations:

\[ P(1) = a_2 + a_1 + a_0 \equiv 1 \pmod{7} \]
\[ P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \]
\[ P(3) = 2a_2 + 3a_1 + a_0 \equiv 4 \pmod{7} \]

\[ 6a_1 + 3a_0 = 2 \pmod{7}, \quad 5a_1 + 4a_0 = 0 \pmod{7} \]

\[ a_1 = 2a_0, \quad a_0 = 2 \pmod{7}, \quad a_1 = 4 \pmod{7}, \quad a_2 = 2 \pmod{7} \]

\[ P(x) = 2x^2 + 4x + 2 \]
Example

Make polynomial with \( P(1) = 1, \ P(2) = 4, \ P(3) = 4. \)
Modulo 7 to accommodate at least 6 packets.

Linear equations:

\[
P(1) = a_2 + a_1 + a_0 \equiv 1 \pmod{7} \\
P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \\
P(3) = 2a_2 + 3a_1 + a_0 \equiv 4 \pmod{7}
\]

\[
6a_1 + 3a_0 = 2 \pmod{7}, \ 5a_1 + 4a_0 = 0 \pmod{7}
\]

\[
a_1 = 2a_0. \ a_0 = 2 \pmod{7} \ a_1 = 4 \pmod{7} \ a_2 = 2 \pmod{7}
\]

\[
P(x) = 2x^2 + 4x + 2
\]

\[
P(1) = 1,
\]
Example

Make polynomial with \( P(1) = 1, \ P(2) = 4, \ P(3) = 4 \).
Modulo 7 to accommodate at least 6 packets.

Linear equations:

\[
\begin{align*}
P(1) &= a_2 + a_1 + a_0 \equiv 1 \pmod{7} \\
P(2) &= 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \\
P(3) &= 2a_2 + 3a_1 + a_0 \equiv 4 \pmod{7}
\end{align*}
\]

\[6a_1 + 3a_0 = 2 \pmod{7}, \ 5a_1 + 4a_0 = 0 \pmod{7}\]

\(a_1 = 2a_0. \ a_0 = 2 \pmod{7} \ a_1 = 4 \pmod{7} \ a_2 = 2 \pmod{7}\)

\[P(x) = 2x^2 + 4x + 2\]

\(P(1) = 1, \ P(2) = 4, \)
Example

Make polynomial with \( P(1) = 1, \ P(2) = 4, \ P(3) = 4 \).

Modulo 7 to accommodate at least 6 packets.

Linear equations:

\[
\begin{align*}
P(1) &= a_2 + a_1 + a_0 \equiv 1 \pmod{7} \\
P(2) &= 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \\
P(3) &= 2a_2 + 3a_1 + a_0 \equiv 4 \pmod{7}
\end{align*}
\]

\[
6a_1 + 3a_0 = 2 \pmod{7}, \quad 5a_1 + 4a_0 = 0 \pmod{7}
\]

\[
a_1 = 2a_0, \quad a_0 = 2 \pmod{7} \quad a_1 = 4 \pmod{7} \quad a_2 = 2 \pmod{7}
\]

\[
P(x) = 2x^2 + 4x + 2
\]

\[P(1) = 1, \ P(2) = 4, \text{ and } P(3) = 4\]
Example

Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$.
Modulo 7 to accommodate at least 6 packets.
Linear equations:

\[
P(1) = a_2 + a_1 + a_0 \equiv 1 \pmod{7}
\]
\[
P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7}
\]
\[
P(3) = 2a_2 + 3a_1 + a_0 \equiv 4 \pmod{7}
\]

\[6a_1 + 3a_0 = 2 \pmod{7}, \quad 5a_1 + 4a_0 = 0 \pmod{7}\]
\[a_1 = 2a_0, \quad a_0 = 2 \pmod{7}, \quad a_1 = 4 \pmod{7}, \quad a_2 = 2 \pmod{7}\]

\[P(x) = 2x^2 + 4x + 2\]
\[P(1) = 1, \quad P(2) = 4, \quad \text{and} \quad P(3) = 4\]
Example

Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$.
Modulo 7 to accommodate at least 6 packets.
Linear equations:

$$P(1) = a_2 + a_1 + a_0 \equiv 1 \pmod{7}$$
$$P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7}$$
$$P(3) = 2a_2 + 3a_1 + a_0 \equiv 4 \pmod{7}$$

$6a_1 + 3a_0 = 2 \pmod{7}$, $5a_1 + 4a_0 = 0 \pmod{7}$

$a_1 = 2a_0$, $a_0 = 2 \pmod{7}$, $a_1 = 4 \pmod{7}$, $a_2 = 2 \pmod{7}$

$P(x) = 2x^2 + 4x + 2$

$P(1) = 1$, $P(2) = 4$, and $P(3) = 4$

Send
Example

Make polynomial with \( P(1) = 1, \ P(2) = 4, \ P(3) = 4. \)
Modulo 7 to accommodate at least 6 packets.
Linear equations:

\[
\begin{align*}
P(1) &= a_2 + a_1 + a_0 \equiv 1 \pmod{7} \\
P(2) &= 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \\
P(3) &= 2a_2 + 3a_1 + a_0 \equiv 4 \pmod{7}
\end{align*}
\]

\[
6a_1 + 3a_0 = 2 \pmod{7}, \quad 5a_1 + 4a_0 = 0 \pmod{7}
\]

\[
a_1 = 2a_0. \quad a_0 = 2 \pmod{7} \quad a_1 = 4 \pmod{7} \quad a_2 = 2 \pmod{7}
\]

\[
P(x) = 2x^2 + 4x + 2
\]

\[
P(1) = 1, \ P(2) = 4, \text{ and } P(3) = 4
\]

Send
Packets: (1, 1), (2, 4), (3, 4), (4, 7), (5, 2), (6, 0)
Example

Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$.
Modulo 7 to accommodate at least 6 packets.
Linear equations:

\[
P(1) = a_2 + a_1 + a_0 \equiv 1 \pmod{7}
\]
\[
P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7}
\]
\[
P(3) = 2a_2 + 3a_1 + a_0 \equiv 4 \pmod{7}
\]

\[
6a_1 + 3a_0 = 2 \pmod{7}, \quad 5a_1 + 4a_0 = 0 \pmod{7}
\]
\[
a_1 = 2a_0. \quad a_0 = 2 \pmod{7} \quad a_1 = 4 \pmod{7} \quad a_2 = 2 \pmod{7}
\]

$P(x) = 2x^2 + 4x + 2$

$P(1) = 1$, $P(2) = 4$, and $P(3) = 4$

Send

Packets: (1, 1), (2, 4), (3, 4), (4, 7), (5, 2), (6, 0)

Notice that packets contain “x-values”.

Bad reception!

Send: (1, 1), (2, 4), (3, 4), (4, 7), (5, 2), (6, 0)
Bad reception!

Send: (1,1), (2,4), (3,4), (4,7), (5,2), (6,0)
Recieve: (1,1) (3,4), (6,0)
Bad reception!

Send: (1, 1), (2, 4), (3, 4), (4, 7), (5, 2), (6, 0)

Receive: (1, 1), (3, 4), (6, 0)
Reconstruct?
Bad reception!

Send: $(1,1), (2,4), (3,4), (4,7), (5,2), (6,0)$

Receive: $(1,1), (3,4), (6,0)$

Reconstruct?

Format: $(i, R(i))$. 
Bad reception!

Send: (1,1), (2,4), (3,4), (4,7), (5,2), (6,0)

Receive: (1,1) (3,4), (6,0)

Reconstruct?

Format: (i, R(i)).

Lagrange or linear equations.
Bad reception!

Send: (1, 1), (2, 4), (3, 4), (4, 7), (5, 2), (6, 0)

Recieve: (1, 1) (3, 4), (6, 0)
   Reconstruct?

Format: \((i, R(i))\).

Lagrange or linear equations.

\[
P(1) = a_2 + a_1 + a_0 \equiv 1 \pmod{7}
\]
Bad reception!

Send: (1, 1), (2, 4), (3, 4), (4, 7), (5, 2), (6, 0)

Receive: (1, 1), (3, 4), (6, 0)

Reconstruct?

Format: \( (i, R(i)) \).

Lagrange or linear equations.

\[
P(1) = a_2 + a_1 + a_0 \equiv 1 \pmod{7}
\]

\[
P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7}
\]
Bad reception!

Send: (1, 1), (2, 4), (3, 4), (4, 7), (5, 2), (6, 0)

Receive: (1, 1) (3, 4), (6, 0)
Reconstruct?

Format: (i, R(i)).

Lagrange or linear equations.

\[ P(1) = a_2 + a_1 + a_0 \equiv 1 \pmod{7} \]
\[ P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \]
\[ P(6) = 2a_2 + 3a_1 + a_0 \equiv 0 \pmod{7} \]
Bad reception!

Send: (1, 1), (2, 4), (3, 4), (4, 7), (5, 2), (6, 0)
Recieve: (1, 1) (3, 4), (6, 0)
Reconstruct?
Format: (i, R(i)).
Lagrange or linear equations.

\[ P(1) = a_2 + a_1 + a_0 \equiv 1 \text{ (mod 7)} \]
\[ P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \text{ (mod 7)} \]
\[ P(6) = 2a_2 + 3a_1 + a_0 \equiv 0 \text{ (mod 7)} \]
Bad reception!

Send: (1,1), (2,4), (3,4), (4,7), (5,2), (6,0)

Receive: (1,1) (3,4), (6,0)
    Reconstruct?

Format: \((i, R(i))\).

Lagrange or linear equations.

\[
\begin{align*}
    P(1) &= a_2 + a_1 + a_0 \equiv 1 \pmod{7} \\
    P(2) &= 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \\
    P(6) &= 2a_2 + 3a_1 + a_0 \equiv 0 \pmod{7}
\end{align*}
\]

Channeling Sahai
Bad reception!

Send: \((1, 1), (2, 4), (3, 4), (4, 7), (5, 2), (6, 0)\)

Receive: \((1,1)\) \((3,4)\), \((6,0)\)

Reconstruct?

Format: \((i, R(i))\).

Lagrange or linear equations.

\[
P(1) = a_2 + a_1 + a_0 \equiv 1 \pmod{7} \\
P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \\
P(6) = 2a_2 + 3a_1 + a_0 \equiv 0 \pmod{7}
\]

Channeling Sahai ...
Bad reception!

Send: \((1, 1), (2, 4), (3, 4), (4, 7), (5, 2), (6, 0)\)

Recieve: \((1, 1), (3, 4), (6, 0)\)

Reconstruct?

Format: \((i, R(i))\).

Lagrange or linear equations.

\[
P(1) = a_2 + a_1 + a_0 \equiv 1 \pmod{7}
\]

\[
P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7}
\]

\[
P(6) = 2a_2 + 3a_1 + a_0 \equiv 0 \pmod{7}
\]

Channeling Sahai ...

\[P(x) = 2x^2 + 4x + 2\]
Bad reception!

Send: (1, 1), (2, 4), (3, 4), (4, 7), (5, 2), (6, 0)

Recieve: (1, 1), (3, 4), (6, 0)
Reconstruct?

Format: (i, R(i)).

Lagrange or linear equations.

\[ P(1) = a_2 + a_1 + a_0 \equiv 1 \pmod{7} \]
\[ P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \]
\[ P(6) = 2a_2 + 3a_1 + a_0 \equiv 0 \pmod{7} \]

Channeling Sahai ...

\[ P(x) = 2x^2 + 4x + 2 \]
Bad reception!

Send: (1, 1), (2, 4), (3, 4), (4, 7), (5, 2), (6, 0)
Recieve: (1, 1) (3, 4), (6, 0)
Reconstruct?
Format: \((i, R(i))\).
Lagrange or linear equations.

\[ P(1) = a_2 + a_1 + a_0 \equiv 1 \pmod{7} \]
\[ P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \]
\[ P(6) = 2a_2 + 3a_1 + a_0 \equiv 0 \pmod{7} \]

Channeling Sahai ...
\[ P(x) = 2x^2 + 4x + 2 \]
Message?
Bad reception!

Send: (1, 1), (2, 4), (3, 4), (4, 7), (5, 2), (6, 0)
Recieve: (1,1) (3,4), (6,0)
Reconstruct?

Format: \((i, R(i))\).

Lagrange or linear equations.

\[
P(1) = a_2 + a_1 + a_0 \equiv 1 \pmod{7}
\]
\[
P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7}
\]
\[
P(6) = 2a_2 + 3a_1 + a_0 \equiv 0 \pmod{7}
\]

Channeling Sahai ...

\[P(x) = 2x^2 + 4x + 2\]

Message? \(P(1) = 1\),
Bad reception!

Send: (1,1), (2,4), (3,4), (4,7), (5,2), (6,0)

Recieve: (1,1) (3,4), (6,0)
  Reconstruct?

Format: (i, R(i)).

Lagrange or linear equations.

\[ P(1) = a_2 + a_1 + a_0 \equiv 1 \pmod{7} \]
\[ P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \]
\[ P(6) = 2a_2 + 3a_1 + a_0 \equiv 0 \pmod{7} \]

Channeling Sahai ...

\[ P(x) = 2x^2 + 4x + 2 \]

Message? \( P(1) = 1, P(2) = 4, \)
Bad reception!

Send: \( (1, 1), (2, 4), (3, 4), (4, 7), (5, 2), (6, 0) \)

Recieve: \( (1, 1), (3, 4), (6, 0) \)
Reconstruct?

Format: \( (i, R(i)) \).

Lagrange or linear equations.

\[
\begin{align*}
P(1) &= a_2 + a_1 + a_0 \equiv 1 \pmod{7} \\P(2) &= 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \\P(6) &= 2a_2 + 3a_1 + a_0 \equiv 0 \pmod{7}
\end{align*}
\]

Channeling Sahai ...

\[P(x) = 2x^2 + 4x + 2\]

Message? \( P(1) = 1, P(2) = 4, P(3) = 4. \)
Questions for Review

You want to encode a secret consisting of 1,4,4.
Questions for Review

You want to encode a secret consisting of 1,4,4.

How big should modulus be?
Questions for Review

You want to encode a secret consisting of 1,4,4.

How big should modulus be?
Larger than 144
Questions for Review

You want to encode a secret consisting of 1,4,4.
How big should modulus be?
Larger than 144 and prime!
Questions for Review

You want to encode a secret consisting of 1,4,4. How big should modulus be?
Larger than 144 and prime!

You want to send a message consisting of packets 1,4,2,3,0 through a noisy channel that loses 3 packets. How big should modulus be?
Larger than 8 and prime!

Send $n$ packets $b$-bit packets, with $k$ errors. Modulus should be larger than $n + k$ and also larger than $2^b$. 
You want to encode a secret consisting of $1,4,4$. How big should modulus be?
   Larger than 144 and prime!

You want to send a message consisting of packets $1,4,2,3,0$ through a noisy channel that loses 3 packets.

Modulus should be larger than $n + k$ and also larger than $2^b$. 
Questions for Review

You want to encode a secret consisting of 1,4,4.
   How big should modulus be?
      Larger than 144 and prime!

You want to send a message consisting of packets 1,4,2,3,0 through a noisy channel that loses 3 packets.
   How big should modulus be?
You want to encode a secret consisting of 1,4,4.

How big should modulus be?
Larger than 144 and prime!

You want to send a message consisting of packets 1,4,2,3,0 through a noisy channel that loses 3 packets.
How big should modulus be?
You want to encode a secret consisting of 1,4,4.

How big should modulus be?
Larger than 144 and prime!

You want to send a message consisting of packets 1,4,2,3,0 through a noisy channel that loses 3 packets.

How big should modulus be?
Larger than 8
You want to encode a secret consisting of 1, 4, 4.
How big should modulus be?
  Larger than 144 and prime!

You want to send a message consisting of packets 1, 4, 2, 3, 0 through a noisy channel that loses 3 packets.
How big should modulus be?
  Larger than 8 and prime!
Questions for Review

You want to encode a secret consisting of 1,4,4.
   How big should modulus be?
     Larger than 144 and prime!

You want to send a message consisting of packets 1,4,2,3,0 through a noisy channel that loses 3 packets.
   How big should modulus be?
     Larger than 8 and prime!

Send $n$ packets $b$-bit packets, with $k$ errors.
You want to encode a secret consisting of 1, 4, 4.
How big should modulus be?
Larger than 144 and prime!

You want to send a message consisting of packets 1, 4, 2, 3, 0 through a noisy channel that loses 3 packets.
How big should modulus be?
Larger than 8 and prime!

Send $n$ packets $b$-bit packets, with $k$ errors.
Modulus should be larger than $n + k$ and also larger than $2^b$. 
Polynomials.
Polynomials.

▶ ..give Secret Sharing.
Polynomials.

- give Secret Sharing.
- give Erasure Codes.
Polynomials.

- give Secret Sharing.
- give Erasure Codes.

Error Correction:
Polynomials.

- give Secret Sharing.
- give Erasure Codes.

**Error Correction:**

Noisy Channel: **corrupts** $k$ packets. (rather than **loss**.)
Polynomials.

- give Secret Sharing.
- give Erasure Codes.

**Error Correction:**

Noisy Channel: corrupts \( k \) packets. (rather than loss.)

Additional Challenge: Finding which packets are corrupt.
Error Correction

Satellite

GPS device
Error Correction

Satellite

3 packet message.

GPS device
Error Correction

Satellite

3 packet message.

GPS device

Corrupts 1 packets.
Error Correction

3 packet message. Send 5.

Corrupts 1 packets.
Error Correction

Satellite

3 packet message. Send 5.

Corrupts 1 packets.
Error Correction

3 packet message. Send 5.

Corrupts 1 packets.
The Scheme.

**Problem:** Communicate $n$ packets $m_1, \ldots, m_n$ on noisy channel that corrupts $\leq k$ packets.
The Scheme.

**Problem:** Communicate $n$ packets $m_1, \ldots, m_n$ on noisy channel that corrupts $\leq k$ packets.

**Reed-Solomon Code:**

1. Make a polynomial, $P(x)$ of degree $n - 1$, that encodes message.
   - $P(1) = m_1, \ldots, P(n) = m_n$.
   - Comment: could encode with packets as coefficients.

2. Send $P(1), \ldots, P(n + 2k)$.

After noisy channel: Receive values $R(1), \ldots, R(n + 2k)$.

Properties:
1. $P(i) = R(i)$ for at least $n + k$ points $i$.
2. $P(x)$ is unique degree $n - 1$ polynomial that contains $\geq n + k$ received points.
The Scheme.

**Problem:** Communicate \( n \) packets \( m_1, \ldots, m_n \) on noisy channel that corrupts \( \leq k \) packets.

**Reed-Solomon Code:**

1. Make a polynomial, \( P(x) \) of degree \( n - 1 \), that encodes message.
   - \( P(1) = m_1, \ldots, P(n) = m_n. \)
The Scheme.

**Problem:** Communicate $n$ packets $m_1, \ldots, m_n$ on noisy channel that corrupts $\leq k$ packets.

**Reed-Solomon Code:**

1. Make a polynomial, $P(x)$ of degree $n - 1$, that encodes message.
   - $P(1) = m_1, \ldots, P(n) = m_n$.
   - Comment: could encode with packets as coefficients.
The Scheme.

**Problem:** Communicate $n$ packets $m_1, \ldots, m_n$ on noisy channel that corrupts $\leq k$ packets.

**Reed-Solomon Code:**

1. Make a polynomial, $P(x)$ of degree $n - 1$, that encodes message.
   - $P(1) = m_1, \ldots, P(n) = m_n$.
   - Comment: could encode with packets as coefficients.

2. Send $P(1), \ldots, P(n + 2k)$. 
The Scheme.

**Problem:** Communicate $n$ packets $m_1, \ldots, m_n$ on noisy channel that corrupts $\leq k$ packets.

**Reed-Solomon Code:**

1. Make a polynomial, $P(x)$ of degree $n-1$, that encodes message.
   - $P(1) = m_1, \ldots, P(n) = m_n$.
   - Comment: could encode with packets as coefficients.

2. Send $P(1), \ldots, P(n+2k)$.

**After noisy channel:** Recieve values $R(1), \ldots, R(n+2k)$. 
The Scheme.

Problem: Communicate \( n \) packets \( m_1, \ldots, m_n \) on noisy channel that corrupts \( \leq k \) packets.

Reed-Solomon Code:

1. Make a polynomial, \( P(x) \) of degree \( n - 1 \), that encodes message.
   - \( P(1) = m_1, \ldots, P(n) = m_n \).
   - Comment: could encode with packets as coefficients.

2. Send \( P(1), \ldots, P(n + 2k) \).

After noisy channel: Recieve values \( R(1), \ldots, R(n + 2k) \).

Properties:
(1) \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
The Scheme.

**Problem:** Communicate $n$ packets $m_1, \ldots, m_n$ on noisy channel that corrupts $\leq k$ packets.

**Reed-Solomon Code:**

1. Make a polynomial, $P(x)$ of degree $n - 1$, that encodes message.
   - $P(1) = m_1, \ldots, P(n) = m_n$.
   - Comment: could encode with packets as coefficients.

2. Send $P(1), \ldots, P(n + 2k)$.

**After noisy channel:** Recieve values $R(1), \ldots, R(n + 2k)$.

**Properties:**

1. $P(i) = R(i)$ for at least $n + k$ points $i$,
2. $P(x)$ is unique degree $n - 1$ polynomial
The Scheme.

**Problem:** Communicate \( n \) packets \( m_1, \ldots, m_n \) on noisy channel that corrupts \( \leq k \) packets.

**Reed-Solomon Code:**

1. Make a polynomial, \( P(x) \) of degree \( n - 1 \), that encodes message.
   - \( P(1) = m_1, \ldots, P(n) = m_n. \)
   - Comment: could encode with packets as coefficients.

2. Send \( P(1), \ldots, P(n + 2k) \).

**After noisy channel:** Recieve values \( R(1), \ldots, R(n + 2k) \).

**Properties:**

1. \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
2. \( P(x) \) is unique degree \( n - 1 \) polynomial that contains \( \geq n + k \) received points.
Properties: proof.

\[ P(x): \text{degree } n - 1 \text{ polynomial.} \]
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n + 2k) \)
Properties: proof.

\[ P(x): \text{degree } n - 1 \text{ polynomial.} \]
Send \[ P(1), \ldots, P(n+2k) \]
Receive \[ R(1), \ldots, R(n+2k) \]
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n + 2k) \)
Receive \( R(1), \ldots, R(n + 2k) \)
At most \( k \) \( i \)’s where \( P(i) \neq R(i) \).
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n + 2k) \)
Receive \( R(1), \ldots, R(n + 2k) \)
At most \( k \) i’s where \( P(i) \neq R(i) \).

**Properties:**
(1) \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n + 2k) \)
Receive \( R(1), \ldots, R(n + 2k) \)
At most \( k \) \( i \)'s where \( P(i) \neq R(i) \).

Properties:
(1) \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
(2) \( P(x) \) is unique degree \( n - 1 \) polynomial
Properties: proof.

$P(x)$: degree $n-1$ polynomial.
Send $P(1), \ldots, P(n+2k)$
Receive $R(1), \ldots, R(n+2k)$
At most $k$ i’s where $P(i) \neq R(i)$.

Properties:
1. $P(i) = R(i)$ for at least $n+k$ points $i$,
2. $P(x)$ is unique degree $n-1$ polynomial
   that contains $\geq n+k$ received points.
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.

Send \( P(1), \ldots, P(n + 2k) \)

Receive \( R(1), \ldots, R(n + 2k) \)

At most \( k \) i’s where \( P(i) \neq R(i) \).

Properties:

1. \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
2. \( P(x) \) is unique degree \( n - 1 \) polynomial
   that contains \( \geq n + k \) received points.

Proof:
Properties: proof.

$P(x)$: degree $n - 1$ polynomial.
Send $P(1), \ldots, P(n + 2k)$
Receive $R(1), \ldots, R(n + 2k)$
At most $k$ i’s where $P(i) \neq R(i)$.

Properties:
(1) $P(i) = R(i)$ for at least $n + k$ points $i$,
(2) $P(x)$ is unique degree $n - 1$ polynomial
    that contains $\geq n + k$ received points.

Proof:
(1) Sure.
Properties: proof.

\[ P(x) : \text{degree } n - 1 \text{ polynomial.} \]
Send \( P(1), \ldots, P(n + 2k) \)
Receive \( R(1), \ldots, R(n + 2k) \)
At most \( k \) i’s where \( P(i) \neq R(i) \).

Properties:
(1) \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
(2) \( P(x) \) is unique degree \( n - 1 \) polynomial
    that contains \( \geq n + k \) received points.

Proof:
(1) Sure. Only \( k \) corruptions.
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n + 2k) \)
Receive \( R(1), \ldots, R(n + 2k) \)
At most \( k \) i’s where \( P(i) \neq R(i) \).

Properties:
(1) \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
(2) \( P(x) \) is unique degree \( n - 1 \) polynomial
   that contains \( \geq n + k \) received points.

Proof:
(1) Sure. Only \( k \) corruptions.
(2) Degree \( n - 1 \) polynomial \( Q(x) \) consistent with \( n + k \) points.
Properties: proof.

$P(x)$: degree $n - 1$ polynomial.
Send $P(1), \ldots, P(n + 2k)$
Receive $R(1), \ldots, R(n + 2k)$
At most $k$ $i$'s where $P(i) \neq R(i)$.

Properties:
(1) $P(i) = R(i)$ for at least $n + k$ points $i$,
(2) $P(x)$ is unique degree $n - 1$ polynomial
    that contains $\geq n + k$ received points.

Proof:
(1) Sure. Only $k$ corruptions.
(2) Degree $n - 1$ polynomial $Q(x)$ consistent with $n + k$ points.
    $Q(x)$ agrees with $R(i)$, $n + k$ times.
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n + 2k) \)
Receive \( R(1), \ldots, R(n + 2k) \)
At most \( k \) i’s where \( P(i) \neq R(i) \).

Properties:
(1) \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
(2) \( P(x) \) is unique degree \( n - 1 \) polynomial
    that contains \( \geq n + k \) received points.

Proof:
(1) Sure. Only \( k \) corruptions.
(2) Degree \( n - 1 \) polynomial \( Q(x) \) consistent with \( n + k \) points.
    \( Q(x) \) agrees with \( R(i), n + k \) times.
\( P(x) \) agrees with \( R(i), n + k \) times.
Properties: proof.

$P(x)$: degree $n - 1$ polynomial.
Send $P(1), \ldots, P(n + 2k)$
Receive $R(1), \ldots, R(n + 2k)$
At most $k$ $i$'s where $P(i) \neq R(i)$.

Properties:
(1) $P(i) = R(i)$ for at least $n + k$ points $i$,
(2) $P(x)$ is unique degree $n - 1$ polynomial
    that contains $\geq n + k$ received points.

Proof:
(1) Sure. Only $k$ corruptions.
(2) Degree $n - 1$ polynomial $Q(x)$ consistent with $n + k$ points.
    $Q(x)$ agrees with $R(i)$, $n + k$ times.
    $P(x)$ agrees with $R(i)$, $n + k$ times.
    Total points contained by both: $2n + 2k$. 
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n+2k) \)
Receive \( R(1), \ldots, R(n+2k) \)
At most \( k \) i’s where \( P(i) \neq R(i) \).

Properties:
(1) \( P(i) = R(i) \) for at least \( n+k \) points \( i \),
(2) \( P(x) \) is unique degree \( n-1 \) polynomial
    that contains \( \geq n+k \) received points.

Proof:
(1) Sure. Only \( k \) corruptions.
(2) Degree \( n-1 \) polynomial \( Q(x) \) consistent with \( n+k \) points.
    \( Q(x) \) agrees with \( R(i), n+k \) times.
    \( P(x) \) agrees with \( R(i), n+k \) times.
    Total points contained by both: \( 2n+2k \). \( P \) Pigeons.
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n + 2k) \)
Receive \( R(1), \ldots, R(n + 2k) \)
At most \( k \) \( i \)'s where \( P(i) \neq R(i) \).

Properties:
1. \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
2. \( P(x) \) is unique degree \( n - 1 \) polynomial that contains \( \geq n + k \) received points.

Proof:
1. Sure. Only \( k \) corruptions.
2. Degree \( n - 1 \) polynomial \( Q(x) \) consistent with \( n + k \) points.
   \( Q(x) \) agrees with \( R(i), n + k \) times.
   \( P(x) \) agrees with \( R(i), n + k \) times.
   Total points contained by both: \( 2n + 2k \). Pigeons.
   Total points to choose from : \( n + 2k \).
**Properties: proof.**

**Proof:**

1. Sure. Only $k$ corruptions.

2. Degree $n-1$ polynomial $Q(x)$ consistent with $n+k$ points.

   - $Q(x)$ agrees with $R(i)$, $n+k$ times.
   - $P(x)$ agrees with $R(i)$, $n+k$ times.

   Total points contained by both: $2n+2k$. $P$ Pigeons.

   Total points to choose from : $n+2k$. $H$ Holes.
Properties: proof.

\[ P(x) \]: degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n+2k) \)
Receive \( R(1), \ldots, R(n+2k) \)
At most \( k \) i’s where \( P(i) \neq R(i) \).

Properties:
(1) \( P(i) = R(i) \) for at least \( n+k \) points \( i \),
(2) \( P(x) \) is unique degree \( n - 1 \) polynomial
    that contains \( \geq n+k \) received points.

Proof:
(1) Sure. Only \( k \) corruptions.
(2) Degree \( n - 1 \) polynomial \( Q(x) \) consistent with \( n+k \) points.
    \( Q(x) \) agrees with \( R(i), n+k \) times.
    \( P(x) \) agrees with \( R(i), n+k \) times.
    Total points contained by both: \( 2n+2k \). \( P \) Pigeons.
    Total points to choose from: \( n+2k \). \( H \) Holes.
    Points contained by both: \( \geq n \).
Properties: proof.

\[ P(x) \]: degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n+2k) \)
Receive \( R(1), \ldots, R(n+2k) \)
At most \( k \) i’s where \( P(i) \neq R(i) \).

Properties:
(1) \( P(i) = R(i) \) for at least \( n+k \) points \( i \),
(2) \( P(x) \) is unique degree \( n - 1 \) polynomial
    that contains \( \geq n+k \) received points.

Proof:
(1) Sure. Only \( k \) corruptions.
(2) Degree \( n - 1 \) polynomial \( Q(x) \) consistent with \( n+k \) points.
    \( Q(x) \) agrees with \( R(i) \), \( n+k \) times.
    \( P(x) \) agrees with \( R(i) \), \( n+k \) times.
    Total points contained by both: \( 2n+2k \). \( P \) Pigeons.
    Total points to choose from : \( n+2k \). \( H \) Holes.
    Points contained by both : \( \geq n \). \( \geq P-H \) Collisions.
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n+2k) \)
Receive \( R(1), \ldots, R(n+2k) \)
At most \( k \) \( i \)'s where \( P(i) \neq R(i) \).

Properties:
(1) \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
(2) \( P(x) \) is unique degree \( n - 1 \) polynomial
    that contains \( \geq n + k \) received points.

Proof:
(1) Sure. Only \( k \) corruptions.
(2) Degree \( n - 1 \) polynomial \( Q(x) \) consistent with \( n + k \) points.
    \( Q(x) \) agrees with \( R(i), n + k \) times.
    \( P(x) \) agrees with \( R(i), n + k \) times.
    Total points contained by both: \( 2n + 2k \). \( P \quad \) Pigeons.
    Total points to choose from : \( n + 2k \). \( H \quad \) Holes.
    Points contained by both : \( \geq n. \quad \geq P - H \quad \) Collisions.
    \( \implies \) \( Q(i) = P(i) \) at \( n \) points.
Properties: proof.

$P(x)$: degree $n - 1$ polynomial.
Send $P(1), \ldots, P(n+2k)$
Receive $R(1), \ldots, R(n+2k)$
At most $k$ i’s where $P(i) \neq R(i)$.

Properties:
(1) $P(i) = R(i)$ for at least $n+k$ points $i$,
(2) $P(x)$ is unique degree $n - 1$ polynomial
    that contains $\geq n+k$ received points.

Proof:
(1) Sure. Only $k$ corruptions.
(2) Degree $n-1$ polynomial $Q(x)$ consistent with $n+k$ points.
    $Q(x)$ agrees with $R(i)$, $n+k$ times.
    $P(x)$ agrees with $R(i)$, $n+k$ times.
    Total points contained by both: $2n+2k$. $P$ Pigeons.
    Total points to choose from : $n+2k$. $H$ Holes.
    Points contained by both : $\geq n$. $\geq P - H$ Collisions.
    $\implies Q(i) = P(i)$ at $n$ points.
    $\implies Q(x) = P(x)$. 
Properties: proof.

\[ P(x) \]: degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n+2k) \)
Receive \( R(1), \ldots, R(n+2k) \)
At most \( k \) i’s where \( P(i) \neq R(i) \).

Properties:
(1) \( P(i) = R(i) \) for at least \( n+k \) points \( i \),
(2) \( P(x) \) is unique degree \( n - 1 \) polynomial
    that contains \( \geq n+k \) received points.

Proof:
(1) Sure. Only \( k \) corruptions.
(2) Degree \( n - 1 \) polynomial \( Q(x) \) consistent with \( n+k \) points.
    \( Q(x) \) agrees with \( R(i) \), \( n+k \) times.
    \( P(x) \) agrees with \( R(i) \), \( n+k \) times.
    Total points contained by both: \( 2n+2k \).  \( P \) Pigeons.
    Total points to choose from : \( n+2k \). \( H \) Holes.
    Points contained by both : \( \geq n \). \( \geq P - H \) Collisions.
    \( \implies Q(i) = P(i) \) at \( n \) points.
    \( \implies Q(x) = P(x) \).
Example.

Message: 3, 0, 6.
Example.

Message: 3, 0, 6.

Reed Solomon Code: $P(x) = x^2 + x + 1 \pmod{7}$ has $P(1) = 3, P(2) = 0, P(3) = 6 \text{ modulo } 7$. 
Example.

Message: 3, 0, 6.

Reed Solomon Code: \( P(x) = x^2 + x + 1 \) (mod 7) has
\( P(1) = 3, P(2) = 0, P(3) = 6 \) modulo 7.

Send: \( P(1) = 3, P(2) = 0, P(3) = 6, \)
Example.

Message: 3, 0, 6.

Reed Solomon Code: \( P(x) = x^2 + x + 1 \) (mod 7) has
\( P(1) = 3, P(2) = 0, P(3) = 6 \) modulo 7.

Send: \( P(1) = 3, P(2) = 0, P(3) = 6, P(4) = 0, P(5) = 3 \).
Example.

Message: 3, 0, 6.

Reed Solomon Code: $P(x) = x^2 + x + 1 \pmod{7}$ has $P(1) = 3, P(2) = 0, P(3) = 6$ modulo 7.

Send: $P(1) = 3, P(2) = 0, P(3) = 6, P(4) = 0, P(5) = 3$.

(Aside: Message in plain text!)
Example.

Message: 3, 0, 6.

Reed Solomon Code: $P(x) = x^2 + x + 1 \pmod{7}$ has $P(1) = 3, P(2) = 0, P(3) = 6$ modulo 7.

Send: $P(1) = 3, P(2) = 0, P(3) = 6, P(4) = 0, P(5) = 3$.

(Aside: Message in plain text!)

Receive $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$. 
Example.

Message: 3, 0, 6.

Reed Solomon Code: \( P(x) = x^2 + x + 1 \) (mod 7) has 
\( P(1) = 3, P(2) = 0, P(3) = 6 \) modulo 7.

Send: \( P(1) = 3, P(2) = 0, P(3) = 6, P(4) = 0, P(5) = 3 \).

(Aside: Message in plain text!)

Receive \( R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3 \).

\( P(i) = R(i) \) for \( n + k = 3 + 1 = 4 \) points.
Slow solution.

**Brute Force:**
For each subset of $n + k$ points
Slow solution.

**Brute Force:**
For each subset of $n + k$ points
  Fit degree $n - 1$ polynomial, $Q(x)$, to $n$ of them.
Brute Force:
For each subset of $n + k$ points
  Fit degree $n - 1$ polynomial, $Q(x)$, to $n$ of them.
  Check if consistent with $n + k$ of the total points.
Slow solution.

**Brute Force:**
For each subset of $n + k$ points
   Fit degree $n - 1$ polynomial, $Q(x)$, to $n$ of them.
   Check if consistent with $n + k$ of the total points.
   If yes, output $Q(x)$. 
Slow solution.

**Brute Force:**
For each subset of $n+k$ points
  - Fit degree $n-1$ polynomial, $Q(x)$, to $n$ of them.
  - Check if consistent with $n+k$ of the total points.
    If yes, output $Q(x)$.

  - For subset of $n+k$ pts where $R(i) = P(i)$,
    method will reconstruct $P(x)$!
Slow solution.

**Brute Force:**
For each subset of $n + k$ points
   Fit degree $n - 1$ polynomial, $Q(x)$, to $n$ of them.
   Check if consistent with $n + k$ of the total points.
   If yes, output $Q(x)$.

- For subset of $n + k$ pts where $R(i) = P(i)$, method will reconstruct $P(x)$!
- For any subset of $n + k$ pts,
Slow solution.

**Brute Force:**
For each subset of $n+k$ points
  Fit degree $n-1$ polynomial, $Q(x)$, to $n$ of them.
  Check if consistent with $n+k$ of the total points.
  If yes, output $Q(x)$.

- For subset of $n+k$ pts where $R(i) = P(i)$, method will reconstruct $P(x)$!
- For any subset of $n+k$ pts,
  1. there is unique degree $n-1$ polynomial $Q(x)$ that fits $n$ of them
Slow solution.

**Brute Force:**
For each subset of $n + k$ points
  - Fit degree $n - 1$ polynomial, $Q(x)$, to $n$ of them.
  - Check if consistent with $n + k$ of the total points.
  - If yes, output $Q(x)$.

- For subset of $n + k$ pts where $R(i) = P(i)$, method will reconstruct $P(x)$!
- For any subset of $n + k$ pts,
  1. there is unique degree $n - 1$ polynomial $Q(x)$ that fits $n$ of them
  2. and where $Q(x)$ is consistent with $n + k$ points
Slow solution.

**Brute Force:**
For each subset of $n+k$ points
  Fit degree $n-1$ polynomial, $Q(x)$, to $n$ of them.
  Check if consistent with $n+k$ of the total points.
  If yes, output $Q(x)$.

- For subset of $n+k$ pts where $R(i) = P(i)$, method will reconstruct $P(x)$!

- For any subset of $n+k$ pts,
  1. there is unique degree $n-1$ polynomial $Q(x)$ that fits $n$ of them
  2. and where $Q(x)$ is consistent with $n+k$ points
     $\implies P(x) = Q(x)$. 
Slow solution.

**Brute Force:**
For each subset of $n+k$ points
   Fit degree $n-1$ polynomial, $Q(x)$, to $n$ of them.
   Check if consistent with $n+k$ of the total points.
   If yes, output $Q(x)$.

- For subset of $n+k$ pts where $R(i) = P(i)$, method will reconstruct $P(x)$!
- For any subset of $n+k$ pts,
  1. there is unique degree $n-1$ polynomial $Q(x)$ that fits $n$ of them
  2. and where $Q(x)$ is consistent with $n+k$ points
     $\implies P(x) = Q(x)$.

Reconstructs $P(x)$ and only $P(x)$!!
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.

All equations:

\[
\begin{align*}
    p_2 + p_1 + p_0 & \equiv 3 \pmod{7} \\
    4p_2 + 2p_1 + p_0 & \equiv 1 \pmod{7} \\
    2p_2 + 3p_1 + p_0 & \equiv 6 \pmod{7} \\
    2p_2 + 4p_1 + p_0 & \equiv 0 \pmod{7} \\
    1p_2 + 5p_1 + p_0 & \equiv 3 \pmod{7}
\end{align*}
\]
Example.

Received \( R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3 \)

Find \( P(x) = p_2 x^2 + p_1 x + p_0 \) that contains \( n + k = 3 + 1 \) points.

All equations:

\[
\begin{align*}
 p_2 + p_1 + p_0 &\equiv 3 \pmod{7} \\
 4p_2 + 2p_1 + p_0 &\equiv 1 \pmod{7} \\
 2p_2 + 3p_1 + p_0 &\equiv 6 \pmod{7} \\
 2p_2 + 4p_1 + p_0 &\equiv 0 \pmod{7} \\
 p_2 + 5p_1 + p_0 &\equiv 3 \pmod{7}
\end{align*}
\]

Assume point 1 is wrong
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.

All equations..

$p_2 + p_1 + p_0 \equiv 3 \pmod{7}$

$4p_2 + 2p_1 + p_0 \equiv 1 \pmod{7}$

$2p_2 + 3p_1 + p_0 \equiv 6 \pmod{7}$

$2p_2 + 4p_1 + p_0 \equiv 0 \pmod{7}$

$p_2 + 5p_1 + p_0 \equiv 3 \pmod{7}$

Assume point 1 is wrong and solve..
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.

All equations..

\[
\begin{align*}
    p_2 + p_1 + p_0 & \equiv 3 \pmod{7} \\
    4p_2 + 2p_1 + p_0 & \equiv 1 \pmod{7} \\
    2p_2 + 3p_1 + p_0 & \equiv 6 \pmod{7} \\
    2p_2 + 4p_1 + p_0 & \equiv 0 \pmod{7} \\
    1p_2 + 5p_1 + p_0 & \equiv 3 \pmod{7}
\end{align*}
\]

Assume point 1 is wrong and solve.. no consistent solution!
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.

All equations:

$$p_2 + p_1 + p_0 \equiv 3 \pmod{7}$$
$$4p_2 + 2p_1 + p_0 \equiv 1 \pmod{7}$$
$$2p_2 + 3p_1 + p_0 \equiv 6 \pmod{7}$$
$$2p_2 + 4p_1 + p_0 \equiv 0 \pmod{7}$$
$$p_2 + 5p_1 + p_0 \equiv 3 \pmod{7}$$

Assume point 1 is wrong and solve. ..no consistent solution!
Assume point 2 is wrong
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.

All equations..

$$p_2 + p_1 + p_0 \equiv 3 \pmod{7}$$
$$4p_2 + 2p_1 + p_0 \equiv 1 \pmod{7}$$
$$2p_2 + 3p_1 + p_0 \equiv 6 \pmod{7}$$
$$2p_2 + 4p_1 + p_0 \equiv 0 \pmod{7}$$
$$p_2 + 5p_1 + p_0 \equiv 3 \pmod{7}$$

Assume point 1 is wrong and solve... no consistent solution!
Assume point 2 is wrong and solve...
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

Find $P(x) = p_2x^2 + p_1x + p_0$ that contains $n + k = 3 + 1$ points.

All equations..

\[
\begin{align*}
    p_2 + p_1 + p_0 & \equiv 3 \pmod{7} \\
    4p_2 + 2p_1 + p_0 & \equiv 1 \pmod{7} \\
    2p_2 + 3p_1 + p_0 & \equiv 6 \pmod{7} \\
    2p_2 + 4p_1 + p_0 & \equiv 0 \pmod{7} \\
    1p_2 + 5p_1 + p_0 & \equiv 3 \pmod{7}
\end{align*}
\]

Assume point 1 is wrong and solve...*no consistent solution!*

Assume point 2 is wrong and solve...*consistent solution!*
In general,

\[ P(x) = p_{n-1}x^{n-1} + \cdots p_0 \] and receive \( R(1), \ldots R(m = n + 2k) \).
In general...

\[ P(x) = p_{n-1}x^{n-1} + \cdots p_0 \] and receive \( R(1), \ldots R(m = n + 2k) \).

\[ p_{n-1} + \cdots p_0 \equiv R(1) \pmod{p} \]
In general.. 

\[ P(x) = p_{n-1}x^{n-1} + \cdots p_0 \] and receive \( R(1), \ldots R(m = n + 2k) \).

\[ p_{n-1} + \cdots p_0 \equiv R(1) \pmod p \]

\[ p_{n-1}2^{n-1} + \cdots p_0 \equiv R(2) \pmod p \]
In general,

\[ P(x) = p_{n-1}x^{n-1} + \cdots + p_0 \] and receive \( R(1), \ldots, R(m = n + 2k) \).

\[
\begin{align*}
p_{n-1} + \cdots + p_0 & \equiv R(1) \pmod{p} \\
p_{n-1}2^{n-1} + \cdots + p_0 & \equiv R(2) \pmod{p} \\
& \quad \vdots \\
p_{n-1}i^{n-1} + \cdots + p_0 & \equiv R(i) \pmod{p} \\
& \quad \vdots \\
p_{n-1}(m)^{n-1} + \cdots + p_0 & \equiv R(m) \pmod{p}
\end{align*}
\]
In general, 

\[ P(x) = p_{n-1}x^{n-1} + \cdots p_0 \] and receive \( R(1), \ldots, R(m = n + 2k) \).

\[
\begin{align*}
  p_{n-1} + \cdots + p_0 & \equiv R(1) \pmod{p} \\
  p_{n-1}2^{n-1} + \cdots + p_0 & \equiv R(2) \pmod{p} \hspace{1cm} \cdots \\
  p_{n-1}i^{n-1} + \cdots + p_0 & \equiv R(i) \pmod{p} \hspace{1cm} \cdots \\
  p_{n-1}(m)^{n-1} + \cdots + p_0 & \equiv R(m) \pmod{p}
\end{align*}
\]

Error!!
In general, 

\[ P(x) = p_{n-1}x^{n-1} + \cdots p_0 \] 

and receive \( R(1), \ldots, R(m = n + 2k) \).

\[
\begin{align*}
    p_{n-1} + \cdots p_0 & \equiv R(1) \pmod{p} \\
p_{n-1}2^{n-1} + \cdots p_0 & \equiv R(2) \pmod{p} \\
    \quad & \quad \quad \vdots \\
p_{n-1}i^{n-1} + \cdots p_0 & \equiv R(i) \pmod{p} \\
    \quad & \quad \quad \vdots \\
p_{n-1}(m)^{n-1} + \cdots p_0 & \equiv R(m) \pmod{p}
\end{align*}
\]

Error!! .... Where???
In general.. 

\[ P(x) = p_{n-1}x^{n-1} + \cdots + p_0 \] and receive \( R(1), \ldots, R(m = n+2k) \).

\[
\begin{align*}
p_{n-1} + \cdots + p_0 & \equiv R(1) \pmod{p} \\
p_{n-1}2^{n-1} + \cdots + p_0 & \equiv R(2) \pmod{p} \\
& \quad \vdots \\
p_{n-1}i^{n-1} + \cdots + p_0 & \equiv R(i) \pmod{p} \\
& \quad \vdots \\
p_{n-1}(m)^{n-1} + \cdots + p_0 & \equiv R(m) \pmod{p}
\end{align*}
\]

Error!! .... Where???
Could be anywhere!!!
In general,

\[ P(x) = p_{n-1}x^{n-1} + \cdots p_0 \] and receive \( R(1), \ldots, R(m = n + 2k) \).

\[
\begin{align*}
    p_{n-1} + \cdots p_0 & \equiv R(1) \pmod{p} \\
p_{n-1}2^{n-1} + \cdots p_0 & \equiv R(2) \pmod{p} \\
    \vdots \\
p_{n-1}i^{n-1} + \cdots p_0 & \equiv R(i) \pmod{p} \\
    \vdots \\
p_{n-1}(m)^{n-1} + \cdots p_0 & \equiv R(m) \pmod{p}
\end{align*}
\]

Error!! .... Where???
Could be anywhere!!! ...so try everywhere.
In general..

\[ P(x) = p_{n-1}x^{n-1} + \cdots + p_0 \] and receive \( R(1), \ldots, R(m = n + 2k) \).

\[
\begin{align*}
 p_{n-1} + \cdots + p_0 & \equiv R(1) \pmod{p} \\
 p_{n-1}2^{n-1} + \cdots + p_0 & \equiv R(2) \pmod{p} \\
 & \quad \vdots \\
 p_{n-1}i^{n-1} + \cdots + p_0 & \equiv R(i) \pmod{p} \\
 & \quad \vdots \\
 p_{n-1}(m)^{n-1} + \cdots + p_0 & \equiv R(m) \pmod{p}
\end{align*}
\]

Error!! .... Where???
Could be anywhere!!! ...so try everywhere.

**Runtime:** \( \binom{n+2k}{k} \) possibilities.
In general, 

\[ P(x) = p_{n-1}x^{n-1} + \cdots p_0 \] and receive \( R(1), \ldots, R(m = n + 2k) \).

\[ p_{n-1} \cdots p_0 \equiv R(1) \pmod{p} \]
\[ p_{n-1}2^{n-1} \cdots p_0 \equiv R(2) \pmod{p} \]
\[ \vdots \]
\[ p_{n-1}i^{n-1} \cdots p_0 \equiv R(i) \pmod{p} \]
\[ \vdots \]
\[ p_{n-1}(m)^{n-1} \cdots p_0 \equiv R(m) \pmod{p} \]

Error!! .... Where???
Could be anywhere!!! ...so try everywhere.

**Runtime:** \( \binom{n+2k}{k} \) possibilities.

Something like \((n/k)^k\) ...Exponential in \(k\).
In general..

\[ P(x) = p_{n-1}x^{n-1} + \cdots p_0 \] and receive \( R(1), \ldots R(m = n + 2k) \).

\[
\begin{align*}
p_{n-1} + \cdots p_0 & \equiv R(1) \pmod{p} \\
p_{n-1}2^{n-1} + \cdots p_0 & \equiv R(2) \pmod{p} \\
& \quad \vdots \\
p_{n-1}i^{n-1} + \cdots p_0 & \equiv R(i) \pmod{p} \\
& \quad \vdots \\
p_{n-1}(m)^{n-1} + \cdots p_0 & \equiv R(m) \pmod{p}
\end{align*}
\]

Error!! .... Where???
Could be anywhere!!! ...so try everywhere.
**Runtime:** \( \binom{n+2k}{k} \) possibilities.

Something like \((n/k)^k\) ...Exponential in \( k \).

How do we find where the bad packets are efficiently?!?!?!
Ditty...
Where oh where
Ditty...

Where oh where can my bad packets be ...
Ditty...

Where oh where can my bad packets be ...
Efficiency

By Welsh-Berlekamp.
Slow solution.

**Brute Force:**
For each subset of $n + k$ points
Slow solution.

**Brute Force:**
For each subset of $n + k$ points
  - Fit degree $n - 1$ polynomial, $Q(x)$, to $n$ of them.
Slow solution.

**Brute Force:**
For each subset of $n + k$ points
  Fit degree $n - 1$ polynomial, $Q(x)$, to $n$ of them.
  Check if consistent with $n + k$ of the total points.
Slow solution.

**Brute Force:**
For each subset of $n+k$ points
  - Fit degree $n-1$ polynomial, $Q(x)$, to $n$ of them.
  - Check if consistent with $n+k$ of the total points.
  - If yes, output $Q(x)$. 
**Brute Force:**
For each subset of \( n + k \) points
- Fit degree \( n - 1 \) polynomial, \( Q(x) \), to \( n \) of them.
- Check if consistent with \( n + k \) of the total points.
- If yes, output \( Q(x) \).

- For subset of \( n + k \) pts where \( R(i) = P(i) \),
  method will reconstruct \( P(x) \)!
Slow solution.

**Brute Force:**
For each subset of $n+k$ points
   Fit degree $n-1$ polynomial, $Q(x)$, to $n$ of them.
   Check if consistent with $n+k$ of the total points.
   If yes, output $Q(x)$.

- For subset of $n+k$ pts where $R(i) = P(i)$, method will reconstruct $P(x)$!
- For any subset of $n+k$ pts,
Slow solution.

**Brute Force:**
For each subset of $n + k$ points
  - Fit degree $n - 1$ polynomial, $Q(x)$, to $n$ of them.
  - Check if consistent with $n + k$ of the total points.
  - If yes, output $Q(x)$.

- For subset of $n + k$ pts where $R(i) = P(i)$, method will reconstruct $P(x)$!
- For any subset of $n + k$ pts,
  1. there is unique degree $n - 1$ polynomial $Q(x)$ that fits $n$ of them
Brute Force:
For each subset of \( n + k \) points
- Fit degree \( n - 1 \) polynomial, \( Q(x) \), to \( n \) of them.
- Check if consistent with \( n + k \) of the total points.
- If yes, output \( Q(x) \).

- For subset of \( n + k \) pts where \( R(i) = P(i) \), method will reconstruct \( P(x) \)!

- For any subset of \( n + k \) pts,
  1. there is unique degree \( n - 1 \) polynomial \( Q(x) \) that fits \( n \) of them
  2. and where \( Q(x) \) is consistent with \( n + k \) points
Slow solution.

**Brute Force:**
For each subset of $n + k$ points
- Fit degree $n - 1$ polynomial, $Q(x)$, to $n$ of them.
- Check if consistent with $n + k$ of the total points.
  - If yes, output $Q(x)$.

- For subset of $n + k$ pts where $R(i) = P(i)$, method will reconstruct $P(x)$!

- For any subset of $n + k$ pts,
  1. there is unique degree $n - 1$ polynomial $Q(x)$ that fits $n$ of them
  2. and where $Q(x)$ is consistent with $n + k$ points
     $$ P(x) = Q(x).$$
Brute Force:
For each subset of $n+k$ points
- Fit degree $n-1$ polynomial, $Q(x)$, to $n$ of them.
- Check if consistent with $n+k$ of the total points.
- If yes, output $Q(x)$.

- For subset of $n+k$ pts where $R(i) = P(i)$, method will reconstruct $P(x)$!

- For any subset of $n+k$ pts,
  1. there is unique degree $n-1$ polynomial $Q(x)$ that fits $n$ of them
  2. and where $Q(x)$ is consistent with $n+k$ points
     $\implies P(x) = Q(x)$.

Reconstructs $P(x)$ and only $P(x)$!!
Where oh where can my bad packets be?

\[ E(1)(p_{n-1} + \cdots p_0) \equiv R(1)E(1) \pmod{p} \]
Where oh where can my bad packets be?

\[ E(1)(p_{n-1} + \cdots p_0) \equiv R(1)E(1) \pmod{p} \]
\[ E(2)(p_{n-1}2^{n-1} + \cdots p_0) \equiv R(2)E(2) \pmod{p} \]
\[ \vdots \]
\[ E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) \equiv R(n+2k)E(m) \pmod{p} \]
Where oh where can my bad packets be?

\[
E(1)(p_{n-1} + \cdots p_0) \equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) \equiv R(2)E(2) \pmod{p} \\
\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) \equiv R(n+2k)E(m) \pmod{p}
\]

**Idea:** Multiply equation \( i \) by 0 if and only if \( P(i) \neq R(i) \).
Where oh where can my **bad packets** be?

\[
E(1)(p_{n-1} + \cdots + p_0) \equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots + p_0) \equiv R(2)E(2) \pmod{p} \\
\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots + p_0) \equiv R(n+2k)E(m) \pmod{p}
\]

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\). All equations satisfied!!!!!
Where oh where can my **bad packets** be?

\[
E(1)(p_{n-1} + \cdots p_0) \equiv R(1)E(1) \pmod{p}
\]
\[
E(2)(p_{n-1}2^{n-1} + \cdots p_0) \equiv R(2)E(2) \pmod{p}
\]
\[
\vdots
\]
\[
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) \equiv R(n+2k)E(m) \pmod{p}
\]

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\).

All equations satisfied!!!!!

But which equations should we multiply by 0?
Where oh where can my **bad packets** be?

\[
E(1)(p_{n-1} + \cdots p_0) \equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) \equiv R(2)E(2) \pmod{p} \\
\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) \equiv R(n+2k)E(m) \pmod{p}
\]

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\). 
All equations satisfied!!!!!

But which equations should we multiply by 0? Where oh where...
Where oh where can my **bad packets** be?

\[ E(1)(\rho_{n-1} + \cdots p_0) \equiv R(1)E(1) \pmod{p} \]
\[ E(2)(\rho_{n-1}2^{n-1} + \cdots p_0) \equiv R(2)E(2) \pmod{p} \]

\[ \vdots \]
\[ E(m)(\rho_{n-1}(m)^{n-1} + \cdots p_0) \equiv R(n+2k)E(m) \pmod{p} \]

**Idea:** Multiply equation \( i \) by 0 if and only if \( P(i) \neq R(i) \).

All equations satisfied!!!!!

But which equations should we multiply by 0? **Where oh where...??**
Where oh where can my bad packets be?

\[
E(1)(\rho_{n-1} + \cdots \rho_0) \equiv R(1)E(1) \pmod{p} \\
E(2)(\rho_{n-1}2^{n-1} + \cdots \rho_0) \equiv R(2)E(2) \pmod{p} \\
\vdots \\
E(m)(\rho_{n-1}(m)^{n-1} + \cdots \rho_0) \equiv R(n + 2k)E(m) \pmod{p}
\]

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\).

All equations satisfied!!!!!

But which equations should we multiply by 0? Where oh where...??

We will use a polynomial!!!
Where oh where can my bad packets be?

\[
E(1)(\rho_{n-1} + \cdots \rho_0) \equiv R(1)E(1) \pmod{p} \\
E(2)(\rho_{n-1}2^{n-1} + \cdots \rho_0) \equiv R(2)E(2) \pmod{p} \\
\vdots \\
E(m)(\rho_{n-1}(m)^{n-1} + \cdots \rho_0) \equiv R(n+2k)E(m) \pmod{p}
\]

**Idea:** Multiply equation \( i \) by 0 if and only if \( P(i) \neq R(i) \). All equations satisfied!!!!!

But which equations should we multiply by 0? *Where oh where...??* We will use a polynomial!!! That we don’t know.
Where oh where can my bad packets be?

\[
E(1)(\rho_{n-1} + \cdots \rho_0) \equiv R(1)E(1) \pmod{p} \\
E(2)(\rho_{n-1}2^{n-1} + \cdots \rho_0) \equiv R(2)E(2) \pmod{p} \\
\vdots \\
E(m)(\rho_{n-1}(m)^{n-1} + \cdots \rho_0) \equiv R(n+2k)E(m) \pmod{p}
\]

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\).
All equations satisfied!!!!!

But which equations should we multiply by 0? Where oh where...??

We will use a polynomial!!! That we don’t know. But can find!
Where oh where can my **bad packets** be?

\[
E(1)\left(p_{n-1} + \cdots + p_0\right) \equiv R(1)E(1) \pmod{p} \\
E(2)\left(p_{n-1}2^{n-1} + \cdots + p_0\right) \equiv R(2)E(2) \pmod{p} \\
\vdots \\
E(m)\left(p_{n-1}(m)^{n-1} + \cdots + p_0\right) \equiv R(n+2k)E(m) \pmod{p}
\]

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\).
All equations satisfied!!!!!

But which equations should we multiply by 0? Where oh where...??

We will use a polynomial!!! That we don’t know. But can find!

Errors at points \(e_1, \ldots, e_k\). (In diagram above, \(e_1 = 2\).)
Where oh where can my bad packets be?

\[
E(1)(\rho_{n-1} + \cdots + \rho_0) \equiv R(1)E(1) \pmod{p}
\]
\[
E(2)(\rho_{n-1}2^{n-1} + \cdots + \rho_0) \equiv R(2)E(2) \pmod{p}
\]
\[
\vdots
\]
\[
E(m)(\rho_{n-1}(m)^{n-1} + \cdots + \rho_0) \equiv R(n+2k)E(m) \pmod{p}
\]

Idea: Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\).
All equations satisfied!!!!!

But which equations should we multiply by 0? Where oh where...??

We will use a polynomial!!! That we don’t know. But can find!

Errors at points \(e_1, \ldots, e_k\). (In diagram above, \(e_1 = 2\).)

Error locator polynomial: \(E(x) = (x - e_1)\)
Where oh where can my bad packets be?

\[
E(1)(\rho_{n-1} + \cdots \rho_0) \equiv R(1)E(1) \pmod{p} \\
E(2)(\rho_{n-1}2^{n-1} + \cdots \rho_0) \equiv R(2)E(2) \pmod{p} \\
\vdots \\
E(m)(\rho_{n-1}(m)^{n-1} + \cdots \rho_0) \equiv R(n+2k)E(m) \pmod{p}
\]

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\).
All equations satisfied!!!!!!

But which equations should we multiply by 0? Where oh where...??

We will use a polynomial!!! That we don’t know. But can find!

Errors at points \(e_1, \ldots, e_k\). (In diagram above, \(e_1 = 2\).)

**Error locator polynomial:** \(E(x) = (x - e_1)(x - e_2)\)
Where oh where can my bad packets be?

\[
E(1)(p_{n-1} + \cdots p_0) \equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) \equiv R(2)E(2) \pmod{p} \\
\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) \equiv R(n+2k)E(m) \pmod{p}
\]

Idea: Multiply equation \( i \) by 0 if and only if \( P(i) \neq R(i) \). All equations satisfied!!!!!!

But which equations should we multiply by 0? Where oh where...??

We will use a polynomial!!! That we don’t know. But can find!

Errors at points \( e_1, \ldots, e_k \). (In diagram above, \( e_1 = 2 \).)

Error locator polynomial: \( E(x) = (x - e_1)(x - e_2) \cdots \)
Where oh where can my **bad packets** be?

\[
E(1)(\rho_{n-1} + \cdots p_0) \equiv R(1)E(1) \pmod{p} \\
E(2)(\rho_{n-1}2^{n-1} + \cdots p_0) \equiv R(2)E(2) \pmod{p} \\
\vdots \\
E(m)(\rho_{n-1}(m)^{n-1} + \cdots p_0) \equiv R(n+2k)E(m) \pmod{p}
\]

**Idea:** Multiply equation \( i \) by 0 if and only if \( P(i) \neq R(i) \).

All equations satisfied!!!!!

But which equations should we multiply by 0? Where oh where...??

We will use a polynomial!!! That we don’t know. But can find!

Errors at points \( e_1, \ldots, e_k \). (In diagram above, \( e_1 = 2 \).

**Error locator polynomial:** \( E(x) = (x - e_1)(x - e_2)\ldots(x - e_k) \).
Where oh where can my bad packets be?

\[
E(1)(\rho_{n-1} + \cdots + \rho_0) \equiv R(1)E(1) \pmod{p} \\
E(2)(\rho_{n-1}2^{n-1} + \cdots + \rho_0) \equiv R(2)E(2) \pmod{p} \\
\vdots \\
E(m)(\rho_{n-1}(m)^{n-1} + \cdots + \rho_0) \equiv R(n+2k)E(m) \pmod{p}
\]

Idea: Multiply equation \( i \) by 0 if and only if \( P(i) \neq R(i) \). All equations satisfied!!!!!

But which equations should we multiply by 0? Where oh where...??

We will use a polynomial!!! That we don’t know. But can find!

Errors at points \( e_1, \ldots, e_k \). (In diagram above, \( e_1 = 2 \).)

Error locator polynomial: \( E(x) = (x - e_1)(x - e_2) \cdots (x - e_k) \).

\( E(i) = 0 \) if and only if \( e_j = i \) for some \( j \).
Where oh where can my bad packets be?

\[
E(1)(p_{n-1} + \cdots p_0) \equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) \equiv R(2)E(2) \pmod{p} \\
\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) \equiv R(n+2k)E(m) \pmod{p}
\]

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\).

All equations satisfied!!!!!!

But which equations should we multiply by 0? Where oh where...??

**We will use a polynomial!!!** That we don’t know. But can find!

Errors at points \(e_1, \ldots, e_k\). (In diagram above, \(e_1 = 2\).

**Error locator polynomial:** \(E(x) = (x - e_1)(x - e_2) \cdots (x - e_k)\).

\(E(i) = 0\) if and only if \(e_j = i\) for some \(j\)

Multiply equations by \(E(\cdot)\).
Where oh where can my bad packets be?

\[ E(1)(p_{n-1} + \cdots p_0) \equiv R(1)E(1) \pmod{p} \]
\[ E(2)(p_{n-1}2^{n-1} + \cdots p_0) \equiv R(2)E(2) \pmod{p} \]
\[ \vdots \]
\[ E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) \equiv R(n+2k)E(m) \pmod{p} \]

**Idea:** Multiply equation \( i \) by 0 if and only if \( P(i) \neq R(i) \).
All equations satisfied!!!!!

But which equations should we multiply by 0? Where oh where...??

We will use a polynomial!!! That we don’t know. But can find!

Errors at points \( e_1, \ldots, e_k \). (In diagram above, \( e_1 = 2 \).)

**Error locator polynomial:** \( E(x) = (x - e_1)(x - e_2) \cdots (x - e_k) \).

\( E(i) = 0 \) if and only if \( e_j = i \) for some \( j \)

Multiply equations by \( E(\cdot) \). (Above \( E(x) = (x-2) \).)
Where oh where can my bad packets be?

\[
E(1)(\rho_{n-1} + \cdots + p_0) \equiv R(1)E(1) \pmod{p} \\
E(2)(\rho_{n-1}2^{n-1} + \cdots + p_0) \equiv R(2)E(2) \pmod{p} \\
\vdots \\
E(m)(\rho_{n-1}(m)^{n-1} + \cdots + p_0) \equiv R(n+2k)E(m) \pmod{p}
\]

Idea: Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\). All equations satisfied!!!!

But which equations should we multiply by 0? Where oh where...??

We will use a polynomial!!! That we don’t know. But can find!

Errors at points \(e_1, \ldots, e_k\). (In diagram above, \(e_1 = 2\).)

Error locator polynomial: \(E(x) = (x - e_1)(x - e_2)\cdots(x - e_k)\).

\(E(i) = 0\) if and only if \(e_j = i\) for some \(j\)

Multiply equations by \(E(\cdot)\). (Above \(E(x) = (x-2)\).)

All equations satisfied!!
Example.

Received \( R(1) = 3 \), \( R(2) = 1 \), \( R(3) = 6 \), \( R(4) = 0 \), \( R(5) = 3 \)
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.

Plugin points...

\[
(1 - e)(p_2 + p_1 + p_0) \equiv (3)(1 - e) \pmod{7}
\]
\[
(2 - e)(4p_2 + 2p_1 + p_0) \equiv (1)(2 - e) \pmod{7}
\]
\[
(3 - e)(2p_2 + 3p_1 + p_0) \equiv (3)(3 - e) \pmod{7}
\]
\[
(4 - e)(2p_2 + 4p_1 + p_0) \equiv (0)(4 - e) \pmod{7}
\]
\[
(5 - e)(4p_2 + 5p_1 + p_0) \equiv (3)(5 - e) \pmod{7}
\]
Example.

Received \( R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3 \)

Find \( P(x) = p_2x^2 + p_1x + p_0 \) that contains \( n + k = 3 + 1 \) points.

Plugin points...

\[
(1 - e)(p_2 + p_1 + p_0) \equiv (3)(1 - e) \pmod{7}
\]

\[
(2 - e)(4p_2 + 2p_1 + p_0) \equiv (1)(2 - e) \pmod{7}
\]

\[
(3 - e)(2p_2 + 3p_1 + p_0) \equiv (3)(3 - e) \pmod{7}
\]

\[
(4 - e)(2p_2 + 4p_1 + p_0) \equiv (0)(4 - e) \pmod{7}
\]

\[
(5 - e)(4p_2 + 5p_1 + p_0) \equiv (3)(5 - e) \pmod{7}
\]

Error locator polynomial: \((x - 2)\).
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

Find $P(x) = p_2x^2 + p_1x + p_0$ that contains $n + k = 3 + 1$ points.

Plugin points...

$(1 - e)(p_2 + p_1 + p_0) \equiv (3)(1 - e) \pmod{7}$

$(2 - e)(4p_2 + 2p_1 + p_0) \equiv (1)(2 - e) \pmod{7}$

$(3 - e)(2p_2 + 3p_1 + p_0) \equiv (3)(3 - e) \pmod{7}$

$(4 - e)(2p_2 + 4p_1 + p_0) \equiv (0)(4 - e) \pmod{7}$

$(5 - e)(4p_2 + 5p_1 + p_0) \equiv (3)(5 - e) \pmod{7}$

Error locator polynomial: $(x - 2)$.

Multiply equation $i$ by $(i - 2)$.
Example.

Received \( R(1) = 3, \ R(2) = 1, \ R(3) = 6, \ R(4) = 0, \ R(5) = 3 \)

Find \( P(x) = p_2 x^2 + p_1 x + p_0 \) that contains \( n + k = 3 + 1 \) points.

Plugin points...

\[
\begin{align*}
(1 - e)(p_2 + p_1 + p_0) &\equiv (3)(1 - e) \pmod{7} \\
(2 - e)(4p_2 + 2p_1 + p_0) &\equiv (1)(2 - e) \pmod{7} \\
(3 - e)(2p_2 + 3p_1 + p_0) &\equiv (3)(3 - e) \pmod{7} \\
(4 - e)(2p_2 + 4p_1 + p_0) &\equiv (0)(4 - e) \pmod{7} \\
(5 - e)(4p_2 + 5p_1 + p_0) &\equiv (3)(5 - e) \pmod{7}
\end{align*}
\]

Error locator polynomial: \( (x - 2) \).

Multiply equation \( i \) by \( (i - 2) \). All equations satisfied!
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.

Plugin points...

$$
(1 - e)(p_2 + p_1 + p_0) \equiv (3)(1 - e) \pmod{7}
$$
$$
(2 - e)(4p_2 + 2p_1 + p_0) \equiv (1)(2 - e) \pmod{7}
$$
$$
(3 - e)(2p_2 + 3p_1 + p_0) \equiv (3)(3 - e) \pmod{7}
$$
$$
(4 - e)(2p_2 + 4p_1 + p_0) \equiv (0)(4 - e) \pmod{7}
$$
$$
(5 - e)(4p_2 + 5p_1 + p_0) \equiv (3)(5 - e) \pmod{7}
$$

Error locator polynomial: $(x - 2)$.

Multiply equation $i$ by $(i - 2)$. All equations satisfied!

But don’t know error locator polynomial!
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$
Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.

Plugin points...

\[
(1 - e)(p_2 + p_1 + p_0) \equiv (3)(1 - e) \pmod{7} \\
(2 - e)(4p_2 + 2p_1 + p_0) \equiv (1)(2 - e) \pmod{7} \\
(3 - e)(2p_2 + 3p_1 + p_0) \equiv (3)(3 - e) \pmod{7} \\
(4 - e)(2p_2 + 4p_1 + p_0) \equiv (0)(4 - e) \pmod{7} \\
(5 - e)(4p_2 + 5p_1 + p_0) \equiv (3)(5 - e) \pmod{7}
\]

Error locator polynomial: $(x - 2)$.

Multiply equation $i$ by $(i - 2)$. All equations satisfied!

But don’t know error locator polynomial! Do know form:
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.

Plugin points...

\[
(1 - e)(p_2 + p_1 + p_0) \equiv (3)(1 - e) \pmod{7} \\
(2 - e)(4p_2 + 2p_1 + p_0) \equiv (1)(2 - e) \pmod{7} \\
(3 - e)(2p_2 + 3p_1 + p_0) \equiv (3)(3 - e) \pmod{7} \\
(4 - e)(2p_2 + 4p_1 + p_0) \equiv (0)(4 - e) \pmod{7} \\
(5 - e)(4p_2 + 5p_1 + p_0) \equiv (3)(5 - e) \pmod{7}
\]

Error locator polynomial: $(x - 2)$.

Multiply equation $i$ by $(i - 2)$. All equations satisfied!

But don’t know error locator polynomial! Do know form: $(x - e)$. 
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$
Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.
Plugin points...

\[
(1 - e)(p_2 + p_1 + p_0) \equiv (3)(1 - e) \pmod{7}
\]
\[
(2 - e)(4p_2 + 2p_1 + p_0) \equiv (1)(2 - e) \pmod{7}
\]
\[
(3 - e)(2p_2 + 3p_1 + p_0) \equiv (3)(3 - e) \pmod{7}
\]
\[
(4 - e)(2p_2 + 4p_1 + p_0) \equiv (0)(4 - e) \pmod{7}
\]
\[
(5 - e)(4p_2 + 5p_1 + p_0) \equiv (3)(5 - e) \pmod{7}
\]

Error locator polynomial: $(x - 2)$.
Multiply equation $i$ by $(i - 2)$. All equations satisfied!

But don’t know error locator polynomial! Do know form: $(x - e)$. 
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

Find $P(x) = p_2x^2 + p_1x + p_0$ that contains $n + k = 3 + 1$ points.

Plugin points...

\[
\begin{align*}
(1 - e)(p_2 + p_1 + p_0) & \equiv (3)(1 - e) \pmod{7} \\
(2 - e)(4p_2 + 2p_1 + p_0) & \equiv (1)(2 - e) \pmod{7} \\
(3 - e)(2p_2 + 3p_1 + p_0) & \equiv (3)(3 - e) \pmod{7} \\
(4 - e)(2p_2 + 4p_1 + p_0) & \equiv (0)(4 - e) \pmod{7} \\
(5 - e)(4p_2 + 5p_1 + p_0) & \equiv (3)(5 - e) \pmod{7}
\end{align*}
\]

Error locator polynomial: $(x - 2)$.

Multiply equation $i$ by $(i - 2)$. All equations satisfied!

But don’t know error locator polynomial! Do know form: $(x - e)$.

4 unknowns ($p_0, p_1, p_2$ and $e$),
Example.

Received \( R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3 \)

Find \( P(x) = p_2 x^2 + p_1 x + p_0 \) that contains \( n + k = 3 + 1 \) points.

Plug in points...

\[
\begin{align*}
(1 - e)(p_2 + p_1 + p_0) & \equiv (3)(1 - e) \pmod{7} \\
(2 - e)(4p_2 + 2p_1 + p_0) & \equiv (1)(2 - e) \pmod{7} \\
(3 - e)(2p_2 + 3p_1 + p_0) & \equiv (3)(3 - e) \pmod{7} \\
(4 - e)(2p_2 + 4p_1 + p_0) & \equiv (0)(4 - e) \pmod{7} \\
(5 - e)(4p_2 + 5p_1 + p_0) & \equiv (3)(5 - e) \pmod{7}
\end{align*}
\]

Error locator polynomial: \( (x - 2) \).

Multiply equation \( i \) by \( (i - 2) \). All equations satisfied!

But don’t know error locator polynomial! Do know form: \( (x - e) \).

4 unknowns \( (p_0, p_1, p_2 \text{ and } e) \), 5 nonlinear equations.
..turn their heads each day,

\[(p_{n-1} + \cdots + p_0) \equiv R(1) \pmod{p}\]
\[
\vdots
\]

\[(p_{n-1}i^{n-1} + \cdots + p_0) \equiv R(i) \pmod{p}\]
\[
\vdots
\]

\[(p_{n-1}(n+2k)^{n-1} + \cdots + p_0) \equiv R(m) \pmod{p}\]
..turn their heads each day,

\[
E(1)(p_{n-1} + \cdots + p_0) \equiv R(1)E(1) \pmod{p} \\
\vdots \\
E(i)(p_{n-1}i^{n-1} + \cdots + p_0) \equiv R(i)E(i) \pmod{p} \\
\vdots \\
E(m)(p_{n-1}(n+2k)^{n-1} + \cdots + p_0) \equiv R(m)E(m) \pmod{p}
\]

...so satisfied, I’m on my way.
..turn their heads each day,

\[ E(1)(p_{n-1} + \cdots + p_0) \equiv R(1)E(1) \pmod{p} \]
\[ E(i)(p_{n-1}i^{n-1} + \cdots + p_0) \equiv R(i)E(i) \pmod{p} \]
\[ E(m)(p_{n-1}(n+2k)^{n-1} + \cdots + p_0) \equiv R(m)E(m) \pmod{p} \]

...so satisfied, I’m on my way.

\[ m = n + 2k \] satisfied equations,
..turn their heads each day,

\[
E(1)(p_{n-1} + \cdots p_0) \equiv R(1)E(1) \pmod{p}
\]
\[
\vdots
\]
\[
E(i)(p_{n-1}i^{n-1} + \cdots p_0) \equiv R(i)E(i) \pmod{p}
\]
\[
\vdots
\]
\[
E(m)(p_{n-1}(n+2k)^{n-1} + \cdots p_0) \equiv R(m)E(m) \pmod{p}
\]

...so satisfied, I’m on my way.

\[m = n + 2k\] satisfied equations, \(n + k\) unknowns.
..turn their heads each day,

\[
E(1)(\rho_{n-1} + \cdots \rho_0) \equiv R(1)E(1) \pmod{p}
\]

\[
\vdots
\]

\[
E(i)(\rho_{n-1}i^{n-1} + \cdots \rho_0) \equiv R(i)E(i) \pmod{p}
\]

\[
\vdots
\]

\[
E(m)(\rho_{n-1}(n+2k)^{n-1} + \cdots \rho_0) \equiv R(m)E(m) \pmod{p}
\]

...so satisfied, I’m on my way.

\[m = n + 2k \text{ satisfied equations, } n + k \text{ unknowns. But nonlinear!}\]
..turn their heads each day,

\[
E(1)(p_{n-1} + \cdots p_0) \equiv R(1)E(1) \pmod{p} \\
\vdots \\
E(i)(p_{n-1}i^{n-1} + \cdots p_0) \equiv R(i)E(i) \pmod{p} \\
\vdots \\
E(m)(p_{n-1}(n+2k)^{n-1} + \cdots p_0) \equiv R(m)E(m) \pmod{p}
\]

...so satisfied, I’m on my way.

\(m = n + 2k\) satisfied equations, \(n + k\) unknowns. But nonlinear!

Let \(Q(x) = E(x)P(x) = a_{n+k-1}x^{n+k-1} + \cdots a_0.\)
..turn their heads each day,

\[ E(1)(p_{n-1} + \cdots + p_0) \equiv R(1)E(1) \pmod{p} \]
\[ \vdots \]
\[ E(i)(p_{n-1}i^{n-1} + \cdots + p_0) \equiv R(i)E(i) \pmod{p} \]
\[ \vdots \]
\[ E(m)(p_{n-1}(n+2k)^{n-1} + \cdots + p_0) \equiv R(m)E(m) \pmod{p} \]

...so satisfied, I’m on my way.

\( m = n + 2k \) satisfied equations, \( n + k \) unknowns. But nonlinear!

Let \( Q(x) = E(x)P(x) = a_{n+k-1}x^{n+k-1} + \cdots + a_0 \).

Equations:

\[ Q(i) = R(i)E(i). \]
..turn their heads each day,

\[ E(1)(p_{n-1} + \cdots p_0) \equiv R(1)E(1) \pmod{p} \]

\vdots

\[ E(i)(p_{n-1}i^{n-1} + \cdots p_0) \equiv R(i)E(i) \pmod{p} \]

\vdots

\[ E(m)(p_{n-1}(n+2k)^{n-1} + \cdots p_0) \equiv R(m)E(m) \pmod{p} \]

...so satisfied, I’m on my way.

\[ m = n + 2k \text{ satisfied equations, } n + k \text{ unknowns. But nonlinear!} \]

Let \( Q(x) = E(x)P(x) = a_{n+k-1}x^{n+k-1} + \cdots a_0 \).

Equations:

\[ Q(i) = R(i)E(i). \]
..turn their heads each day,

\[ E(1)(p_{n-1} + \cdots + p_0) \equiv R(1)E(1) \pmod{p} \]

\[ \vdots \]

\[ E(i)(p_{n-1}i^{n-1} + \cdots + p_0) \equiv R(i)E(i) \pmod{p} \]

\[ \vdots \]

\[ E(m)(p_{n-1}(n+2k)^{n-1} + \cdots + p_0) \equiv R(m)E(m) \pmod{p} \]

...so satisfied, I’m on my way.

\[ m = n + 2k \] satisfied equations, \( n + k \) unknowns. But nonlinear!

Let \( Q(x) = E(x)P(x) = a_{n+k-1}x^{n+k-1} + \cdots a_0 \).

Equations:

\[ Q(i) = R(i)E(i). \]
..turn their heads each day,

\[
E(1)(p_{n-1} + \cdots p_0) \equiv R(1)E(1) \pmod{p}
\]
\[
\vdots
\]
\[
E(i)(p_{n-1}i^{n-1} + \cdots p_0) \equiv R(i)E(i) \pmod{p}
\]
\[
\vdots
\]
\[
E(m)(p_{n-1}(n + 2k)^{n-1} + \cdots p_0) \equiv R(m)E(m) \pmod{p}
\]

...so satisfied, I’m on my way.

\(m = n + 2k\) satisfied equations, \(n + k\) unknowns. But nonlinear!

Let \(Q(x) = E(x)P(x) = a_{n+k-1}x^{n+k-1} + \cdots a_0\).

Equations:

\[Q(i) = R(i)E(i)\]

and linear in \(a_i\) and coefficients of \(E(x)\)!
Finding $Q(x)$ and $E(x)$?
Finding $Q(x)$ and $E(x)$?

- $E(x)$ has degree $k$
Finding $Q(x)$ and $E(x)$?

- $E(x)$ has degree $k$ ...

\[ E(x) = x^k + b_{k-1}x^{k-1} \cdots b_0. \]
Finding $Q(x)$ and $E(x)$?

- $E(x)$ has degree $k$ ...

$$E(x) = x^k + b_{k-1} x^{k-1} \cdots b_0.$$  

$\implies$ $k$ (unknown) coefficients.
Finding $Q(x)$ and $E(x)$?

- $E(x)$ has degree $k$ ...

$$E(x) = x^k + b_{k-1}x^{k-1} \cdots b_0.$$ 

$\implies k$ (unknown) coefficients. Leading coefficient is 1.
Finding $Q(x)$ and $E(x)$?

- $E(x)$ has degree $k$ ...

$$E(x) = x^k + b_{k-1}x^{k-1} \cdots b_0.$$  

$\implies k$ (unknown) coefficients. Leading coefficient is 1.

- $Q(x) = P(x)E(x)$ has degree $n + k - 1$
Finding $Q(x)$ and $E(x)$?

- $E(x)$ has degree $k$ ...

$$E(x) = x^k + b_{k-1}x^{k-1} \cdots b_0.$$  

$\implies$ $k$ (unknown) coefficients. Leading coefficient is 1.

- $Q(x) = P(x)E(x)$ has degree $n + k - 1$ ...

$$Q(x) = a_{n+k-1}x^{n+k-1} + a_{n+k-2}x^{n+k-2} + \cdots a_0$$
Finding $Q(x)$ and $E(x)$?

- $E(x)$ has degree $k$ ...

\[ E(x) = x^k + b_{k-1} x^{k-1} \cdots b_0. \]

\[ \implies k \text{ (unknown) coefficients. Leading coefficient is } 1. \]

- $Q(x) = P(x)E(x)$ has degree $n + k - 1$ ...

\[ Q(x) = a_{n+k-1} x^{n+k-1} + a_{n+k-2} x^{n+k-2} + \cdots a_0 \]

\[ \implies n + k \text{ (unknown) coefficients.} \]
Finding $Q(x)$ and $E(x)$?

- $E(x)$ has degree $k$ ...

$$E(x) = x^k + b_{k-1}x^{k-1} \cdots b_0.$$ 

$\implies k$ (unknown) coefficients. Leading coefficient is 1.

- $Q(x) = P(x)E(x)$ has degree $n + k - 1$ ...

$$Q(x) = a_{n+k-1}x^{n+k-1} + a_{n+k-2}x^{n+k-2} + \cdots a_0$$ 

$\implies n + k$ (unknown) coefficients.

Number of unknown coefficients:
Finding $Q(x)$ and $E(x)$?

- $E(x)$ has degree $k$ ...

\[ E(x) = x^k + b_{k-1}x^{k-1} \cdots b_0. \]

\[ \implies k \text{ (unknown) coefficients. Leading coefficient is 1.} \]

- $Q(x) = P(x)E(x)$ has degree $n + k - 1$ ...

\[ Q(x) = a_{n+k-1}x^{n+k-1} + a_{n+k-2}x^{n+k-2} + \cdots a_0 \]

\[ \implies n + k \text{ (unknown) coefficients.} \]

Number of unknown coefficients: $n + 2k$. 
Solving for $Q(x)$ and $E(x)$...

For all points $1, \ldots, i, n+2k = m$,

$$Q(i) = R(i)E(i) \pmod{p}$$
Solving for $Q(x)$ and $E(x)$...

For all points $1, \ldots, i, n+2k = m$,

$$Q(i) = R(i)E(i) \pmod{p}$$

Gives $n + 2k$ linear equations.
Solving for $Q(x)$ and $E(x)$...

For all points $1, \ldots, i, n + 2k = m,$

$$Q(i) = R(i)E(i) \pmod{p}$$

Gives $n + 2k$ linear equations.

$$a_{n+k-1} + \ldots a_0 \equiv R(1)(1 + b_{k-1} \ldots b_0) \pmod{p}$$
Solving for $Q(x)$ and $E(x)$...

For all points $1, \ldots, i, n + 2k = m$,

$$Q(i) = R(i)E(i) \pmod{p}$$

Gives $n + 2k$ linear equations.

$$a_{n+k-1} + \ldots a_0 \equiv R(1)(1 + b_{k-1} \cdots b_0) \pmod{p}$$

$$a_{n+k-1}(2)^{n+k-1} + \ldots a_0 \equiv R(2)((2)^k + b_{k-1}(2)^{k-1} \cdots b_0) \pmod{p}$$

\[ \vdots \]
For all points 1, \ldots, i, n + 2k = m,

\[ Q(i) = R(i)E(i) \pmod{p} \]

Gives \( n + 2k \) linear equations.

\[
\begin{align*}
a_{n+k-1} + \cdots + a_0 & \equiv R(1)(1 + b_{k-1} \cdots b_0) \pmod{p} \\
a_{n+k-1}(2^{n+k-1}) + \cdots + a_0 & \equiv R(2)((2)^k + b_{k-1}(2)^{k-1} \cdots b_0) \pmod{p} \\
& \vdots \\
a_{n+k-1}(m^{n+k-1}) + \cdots + a_0 & \equiv R(m)((m)^k + b_{k-1}(m)^{k-1} \cdots b_0) \pmod{p}
\end{align*}
\]
Solving for $Q(x)$ and $E(x)$...

For all points $1, \ldots, i, n + 2k = m$,

$$Q(i) = R(i)E(i) \pmod{p}$$

Gives $n + 2k$ linear equations.

$$a_{n+k-1} + \ldots a_0 \equiv R(1)(1 + b_{k-1} \cdots b_0) \pmod{p}$$

$$a_{n+k-1}(2)^{n+k-1} + \ldots a_0 \equiv R(2)((2)^k + b_{k-1}(2)^{k-1} \cdots b_0) \pmod{p}$$

$$\vdots$$

$$a_{n+k-1}(m)^{n+k-1} + \ldots a_0 \equiv R(m)((m)^k + b_{k-1}(m)^{k-1} \cdots b_0) \pmod{p}$$

..and $n + 2k$ unknown coefficients of $Q(x)$ and $E(x)$!
Solving for $Q(x)$ and $E(x)$...

For all points $1, \ldots, i, n+2k = m$,

$$Q(i) = R(i)E(i) \pmod{p}$$

Gives $n+2k$ linear equations.

\begin{align*}
a_{n+k-1} + \ldots + a_0 & \equiv R(1)(1 + b_{k-1} \ldots b_0) \pmod{p} \\ a_{n+k-1}(2)^{n+k-1} + \ldots + a_0 & \equiv R(2)((2)^k + b_{k-1}(2)^{k-1} \ldots b_0) \pmod{p} \\
& \quad \vdots \\
a_{n+k-1}(m)^{n+k-1} + \ldots + a_0 & \equiv R(m)((m)^k + b_{k-1}(m)^{k-1} \ldots b_0) \pmod{p}
\end{align*}

..and $n+2k$ unknown coefficients of $Q(x)$ and $E(x)$!

Solve for coefficients of $Q(x)$ and $E(x)$. 
Solving for $Q(x)$ and $E(x)$...and $P(x)$

For all points $1, \ldots, i, n+2k = m$,

$$Q(i) = R(i)E(i) \pmod{p}$$

Gives $n+2k$ linear equations.

$$a_{n+k-1} + \ldots + a_0 \equiv R(1)(1 + b_{k-1} \ldots b_0) \pmod{p}$$
$$a_{n+k-1}(2)^{n+k-1} + \ldots + a_0 \equiv R(2)((2)^k + b_{k-1}(2)^{k-1} \ldots b_0) \pmod{p}$$
$$\vdots$$
$$a_{n+k-1}(m)^{n+k-1} + \ldots + a_0 \equiv R(m)((m)^k + b_{k-1}(m)^{k-1} \ldots b_0) \pmod{p}$$

.. and $n+2k$ unknown coefficients of $Q(x)$ and $E(x)$!

Solve for coefficients of $Q(x)$ and $E(x)$.

Find $P(x) = Q(x)/E(x)$. 
Solving for $Q(x)$ and $E(x)$...and $P(x)$

For all points $1, \ldots, i, n+2k = m,$

$$Q(i) = R(i)E(i) \pmod{p}$$

Gives $n+2k$ linear equations.

$$a_{n+k-1} + \ldots a_0 \equiv R(1)(1 + b_{k-1} \cdots b_0) \pmod{p}$$

$$a_{n+k-1}(2)^{n+k-1} + \ldots a_0 \equiv R(2)((2)^k + b_{k-1}(2)^{k-1} \cdots b_0) \pmod{p}$$

$$\vdots$$

$$a_{n+k-1}(m)^{n+k-1} + \ldots a_0 \equiv R(m)((m)^k + b_{k-1}(m)^{k-1} \cdots b_0) \pmod{p}$$

..and $n+2k$ unknown coefficients of $Q(x)$ and $E(x)$!

Solve for coefficients of $Q(x)$ and $E(x)$.

Find $P(x) = Q(x)/E(x)$. 
Solving for $Q(x)$ and $E(x)$...and $P(x)$

For all points $1, \ldots, i, n+2k = m$,

$$Q(i) = R(i)E(i) \pmod{p}$$

Gives $n+2k$ linear equations.

$$a_{n+k-1} + \ldots a_0 \equiv R(1)(1 + b_{k-1} \cdots b_0) \pmod{p}$$
$$a_{n+k-1}(2)^{n+k-1} + \ldots a_0 \equiv R(2)((2)^k + b_{k-1}(2)^{k-1} \cdots b_0) \pmod{p}$$

$$\vdots$$

$$a_{n+k-1}(m)^{n+k-1} + \ldots a_0 \equiv R(m)((m)^k + b_{k-1}(m)^{k-1} \cdots b_0) \pmod{p}$$

..and $n+2k$ unknown coefficients of $Q(x)$ and $E(x)$!

Solve for coefficients of $Q(x)$ and $E(x)$.

Find $P(x) = Q(x)/E(x)$. 
Solving for $Q(x)$ and $E(x)$...and $P(x)$

For all points 1, . . . , $i, n+2k = m,$

$$Q(i) = R(i)E(i) \pmod{p}$$

Gives $n+2k$ linear equations.

1. \(a_{n+k-1} + \ldots a_0 \equiv R(1)(1 + b_{k-1} \ldots b_0) \pmod{p}\)
2. \(a_{n+k-1}(2)^{n+k-1} + \ldots a_0 \equiv R(2)((2)^k + b_{k-1}(2)^{k-1} \ldots b_0) \pmod{p}\)
3. \[\ldots\]
4. \(a_{n+k-1}(m)^{n+k-1} + \ldots a_0 \equiv R(m)((m)^k + b_{k-1}(m)^{k-1} \ldots b_0) \pmod{p}\)

..and $n+2k$ unknown coefficients of $Q(x)$ and $E(x)$!

Solve for coefficients of $Q(x)$ and $E(x)$.

Find $P(x) = Q(x)/E(x)$. 
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$
Example.

Received \( R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3 \)

\[ Q(x) = E(x)P(x) = a_3x^3 + a_2x^2 + a_1x + a_0 \]
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$

$Q(x) = E(x)P(x) = a_3 x^3 + a_2 x^2 + a_1 x + a_0$

$E(x) = x - b_0$
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$

$Q(x) = E(x)P(x) = a_3x^3 + a_2x^2 + a_1x + a_0$

$E(x) = x - b_0$

$Q(i) = R(i)E(i)$. 

Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$

$Q(x) = E(x)P(x) = a_3 x^3 + a_2 x^2 + a_1 x + a_0$

$E(x) = x - b_0$

$Q(i) = R(i)E(i)$.

\[ a_3 + a_2 + a_1 + a_0 \equiv 3(1 - b_0) \pmod{7} \]
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

$Q(x) = E(x)P(x) = a_3x^3 + a_2x^2 + a_1x + a_0$

$E(x) = x - b_0$

$Q(i) = R(i)E(i)$.

\[
\begin{align*}
    a_3 + a_2 + a_1 + a_0 & \equiv 3(1 - b_0) \pmod{7} \\
    a_3 + 4a_2 + 2a_1 + a_0 & \equiv 1(2 - b_0) \pmod{7}
\end{align*}
\]
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$

$Q(x) = E(x)P(x) = a_3 x^3 + a_2 x^2 + a_1 x + a_0$

$E(x) = x - b_0$

$Q(i) = R(i)E(i)$.

\[
\begin{align*}
    a_3 + a_2 + a_1 + a_0 & \equiv 3(1 - b_0) \quad \text{(mod 7)} \\
    a_3 + 4a_2 + 2a_1 + a_0 & \equiv 1(2 - b_0) \quad \text{(mod 7)} \\
    6a_3 + 2a_2 + 3a_1 + a_0 & \equiv 6(3 - b_0) \quad \text{(mod 7)} \\
    a_3 + 2a_2 + 4a_1 + a_0 & \equiv 0(4 - b_0) \quad \text{(mod 7)} \\
    6a_3 + 4a_2 + 5a_1 + a_0 & \equiv 3(5 - b_0) \quad \text{(mod 7)}
\end{align*}
\]
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$

$Q(x) = E(x)P(x) = a_3x^3 + a_2x^2 + a_1x + a_0$

$E(x) = x - b_0$

$Q(i) = R(i)E(i)$. 

\[
\begin{align*}
  a_3 + a_2 + a_1 + a_0 & \equiv 3(1 - b_0) \pmod{7} \\
  a_3 + 4a_2 + 2a_1 + a_0 & \equiv 1(2 - b_0) \pmod{7} \\
  6a_3 + 2a_2 + 3a_1 + a_0 & \equiv 6(3 - b_0) \pmod{7} \\
  a_3 + 2a_2 + 4a_1 + a_0 & \equiv 0(4 - b_0) \pmod{7} \\
  6a_3 + 4a_2 + 5a_1 + a_0 & \equiv 3(5 - b_0) \pmod{7}
\end{align*}
\]

$a_3 = 1, a_2 = 6, a_1 = 6, a_0 = 5$ and $b_0 = 2$. 
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

$Q(x) = E(x)P(x) = a_3x^3 + a_2x^2 + a_1x + a_0$

$E(x) = x - b_0$

$Q(i) = R(i)E(i)$.

\[
\begin{align*}
    a_3 + a_2 + a_1 + a_0 & \equiv 3(1 - b_0) \pmod{7} \\
    a_3 + 4a_2 + 2a_1 + a_0 & \equiv 1(2 - b_0) \pmod{7} \\
    6a_3 + 2a_2 + 3a_1 + a_0 & \equiv 6(3 - b_0) \pmod{7} \\
    a_3 + 2a_2 + 4a_1 + a_0 & \equiv 0(4 - b_0) \pmod{7} \\
    6a_3 + 4a_2 + 5a_1 + a_0 & \equiv 3(5 - b_0) \pmod{7}
\end{align*}
\]

$a_3 = 1$, $a_2 = 6$, $a_1 = 6$, $a_0 = 5$ and $b_0 = 2$.

$Q(x) = x^3 + 6x^2 + 6x + 5$. 
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

$Q(x) = E(x)P(x) = a_3x^3 + a_2x^2 + a_1x + a_0$

$E(x) = x - b_0$

$Q(i) = R(i)E(i)$.

\[
\begin{align*}
    a_3 + a_2 + a_1 + a_0 & \equiv 3(1 - b_0) \pmod{7} \\
    a_3 + 4a_2 + 2a_1 + a_0 & \equiv 1(2 - b_0) \pmod{7} \\
    6a_3 + 2a_2 + 3a_1 + a_0 & \equiv 6(3 - b_0) \pmod{7} \\
    a_3 + 2a_2 + 4a_1 + a_0 & \equiv 0(4 - b_0) \pmod{7} \\
    6a_3 + 4a_2 + 5a_1 + a_0 & \equiv 3(5 - b_0) \pmod{7}
\end{align*}
\]

$a_3 = 1$, $a_2 = 6$, $a_1 = 6$, $a_0 = 5$ and $b_0 = 2$. 

$Q(x) = x^3 + 6x^2 + 6x + 5$. 

$E(x) = x - 2$. 
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{c}
\text{x - 2) x^3 + 6x^2 + 6x + 5} \\
\hline
x^3 - 2x^2 \\
\hline
6x^2 + 6x \\
\hline
6x + 5 \\
\hline
0
\end{array}
\]

Message is

\[ P(x) = x^2 + x + 1 \]

\[ P(1) = 3, \quad P(2) = 0, \quad P(3) = 6. \]

What is \( x - 2 \)?

Except at \( x = 2 \)?

Hole there?
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{r}
  1 \\
  \hline
  x - 2 \\
\end{array}
\]
\[ x^3 + 6x^2 + 6x + 5 \]
\[ \overline{\underline{x^3 - 2x^2}} \]
\[ x^2 + 6x + 5 \]
\[ \overline{\underline{x^2 - 2x}} \]
\[ x + 5 \]
\[ \overline{\underline{x - 2}} \]
\[ 0 \]

\[ P(x) = x^2 + x + 1 \]

Message is

\[ P(1) = 3, \]
\[ P(2) = 0, \]
\[ P(3) = 6. \]

What is \( x - 2 \)?

Except at \( x = 2? \) Hole there?
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{c}
1 \ x^2 \\
\hline
x - 2 & x^3 + 6x^2 + 6x + 5 \\
\hline
x^3 - 2x^2 & \\
\hline
1 \ x^2 + 6x + 5
\end{array}
\]
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{r}
1 & x^2 & + & 1 & x \\
\hline
x - 2 & ) & x^3 & + & 6 & x^2 & + & 6 & x & + & 5 \\
& x^3 & - & 2 & x^2 \\
\hline
& 1 & x^2 & + & 6 & x & + & 5 \\
& 1 & x^2 & - & 2 & x \\
\end{array}
\]

Message is \( P(1) = 3, P(2) = 0, P(3) = 6. \)

What is \( x - 2 \)? Except at \( x = 2 \)? Hole there?
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{c}
1 \ x^2 + 1 \ x \\
\hline
x - 2 \ ) \ x^3 + 6 \ x^2 + 6 \ x + 5 \\
x^3 - 2 \ x^2 \\
\hline
1 \ x^2 + 6 \ x + 5 \\
1 \ x^2 - 2 \ x \\
\hline
x + 5
\end{array}
\]

Message is

\[ P(1) = 3, \quad P(2) = 0, \quad P(3) = 6. \]

What is \( x - 2 \) except at \( x = 2 \)?

Hole there?
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{cccc}
1 & x^2 & + & 1 & x & + & 1 \\
\hline
x - 2 & \) & x^3 & + & 6 & x^2 & + & 6 & x & + & 5 \\
& x^3 & - & 2 & x^2 & \\
\hline
1 & x^2 & + & 6 & x & + & 5 \\
1 & x^2 & - & 2 & x \\
\hline
x & + & 5 \\
x & - & 2
\end{array}
\]
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{c}
1 \ x^2 \ + \ 1 \ x \ + \ 1 \\
\hline
x - 2 \ ) \ x^3 \ + \ 6 \ x^2 \ + \ 6 \ x \ + \ 5 \\
x^3 \ - \ 2 \ x^2 \\
\hline
1 \ x^2 \ + \ 6 \ x \ + \ 5 \\
1 \ x^2 \ - \ 2 \ x \\
\hline
x \ + \ 5 \\
x \ - \ 2 \\
\hline
0
\end{array}
\]

Message is
\[ P(x) = x^2 + x + 1 \]

\[
\begin{align*}
P(1) &= 3, \\
P(2) &= 0, \\
P(3) &= 6.
\end{align*}
\]

What is \( x - 2 \) except at \( x = 2 \)?

Hole there?
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{l}
1 \, x^2 + 1 \, x + 1 \\
\hline
x - 2 ) x^3 + 6 \, x^2 + 6 \, x + 5 \\
\quad \quad \quad \quad \quad \quad x^3 - 2 \, x^2 \\
\hline
\quad \quad \quad \quad \quad \quad 1 \, x^2 + 6 \, x + 5 \\
\quad \quad \quad \quad \quad \quad \quad \quad 1 \, x^2 - 2 \, x \\
\hline
\quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad x + 5 \\
\quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad x - 2 \\
\quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad 0 \\
\end{array}
\]

\[ P(x) = x^2 + x + 1 \]
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{r}
1 & x^2 & + & 1 & x & + & 1 \\
\hline
x - 2 & ) & x^3 & + & 6x^2 & + & 6x & + & 5 \\
\quad x^3 & - & 2x^2 & & & & & & \\
\hline
& & 1x^2 & + & 6x & + & 5 \\
\hline
1 & x^2 & - & 2x & & & & & \\
\hline
& & x & + & 5 & & & & \\
& & x & - & 2 & & & & \\
& & - & & & & & & \\
& & 0 & & & & & & \\
\end{array}
\]

\[ P(x) = x^2 + x + 1 \]
Message is \( P(1) = 3, P(2) = 0, P(3) = 6. \)
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[ \begin{array}{c}
1 \ x^2 + 1 \ x + 1 \\
\hline
x - 2 \ ) \ x^3 + 6 \ x^2 + 6 \ x + 5 \\
\hline
\end{array} \]

\[ \begin{array}{c}
x^3 - 2 \ x^2 \\
\hline
\end{array} \]

\[ \begin{array}{c}
1 \ x^2 + 6 \ x + 5 \\
\hline
1 \ x^2 - 2 \ x \\
\hline
\end{array} \]

\[ \begin{array}{c}
x^2 + 5 \\
\hline
\end{array} \]

\[ \begin{array}{c}
x + 5 \\
\hline
\end{array} \]

\[ \begin{array}{c}
x - 2 \\
\hline
\end{array} \]

\[ 0 \]

\[ P(x) = x^2 + x + 1 \]

Message is \( P(1) = 3, P(2) = 0, P(3) = 6. \)

What is \( \frac{x-2}{x-2} \)?
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{r}
  1 \ x^2 + 1 \ x + 1 \\
\hline
  x - 2 \ ) \ x^3 + 6 \ x^2 + 6 \ x + 5 \\
  \hline
  x^3 - 2 \ x^2 \\
\hline
  1 \ x^2 + 6 \ x + 5 \\
  1 \ x^2 - 2 \ x \\
\hline
  x + 5 \\
  x - 2 \\
\hline
  0
\end{array}
\]

\[ P(x) = x^2 + x + 1 \]

Message is \( P(1) = 3, P(2) = 0, P(3) = 6. \)

What is \( \frac{x-2}{x-2} \)? 1
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5 \]
\[ E(x) = x - 2 \]

\[
\begin{array}{r}
1 & x^2 & + & 1 & x & + & 1 \\
\hline
x - 2 & ) & x^3 & + & 6x^2 & + & 6x & + & 5 \\
& x^3 & - & 2x^2 & & & & & \\
& & & & 1x^2 & + & 6x & + & 5 \\
& 1x^2 & - & 2x & & & & & \\
& & & & & & x & + & 5 \\
& & & & & & x & - & 2 \\
& & & & & & & & 0 
\end{array}
\]

\[ P(x) = x^2 + x + 1 \]

Message is \( P(1) = 3, P(2) = 0, P(3) = 6 \).

What is \( \frac{x-2}{x-2} \)? 1
Except at \( x = 2 \)?
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{c}
1 \ x^2 + 1 \ x + 1 \\
\hline
x - 2 \ ) \ x^3 + 6 \ x^2 + 6 \ x + 5 \\
x^3 - 2 \ x^2 \\
\hline
1 \ x^2 + 6 \ x + 5 \\
1 \ x^2 - 2 \ x \\
\hline
x + 5 \\
x - 2 \\
\hline
0
\end{array}
\]

\[ P(x) = x^2 + x + 1 \]
Message is \( P(1) = 3, P(2) = 0, P(3) = 6. \)

What is \( \frac{x-2}{x-2} ? 1 \) 

Except at \( x = 2 ? \) Hole there?
Error Correction: Berlekamp-Welsh

Message: $m_1, \ldots, m_n$.

**Sender:**

1. Form degree $n-1$ polynomial $P(x)$ where $P(i) = m_i$.
2. Send $P(1), \ldots, P(n+2k)$.

**Receiver:**

1. Receive $R(1), \ldots, R(n+2k)$.
2. Solve $n+2k$ equations, $Q(i) = E(i)R(i)$ to find $Q(x) = E(x)P(x)$ and $E(x)$.
3. Compute $P(x) = Q(x)/E(x)$.
4. Compute $P(1), \ldots, P(n)$. 
Check your understanding.

You have error locator polynomial!
Check your understanding.

You have error locator polynomial!
Where oh where have my packets gone wrong?
Check your understanding.

You have error locator polynomial!
Where oh where have my packets gone wrong?
Factor?
Check your understanding.

You have error locator polynomial!
Where oh where have my packets gone wrong?
Factor? Sure.
Check your understanding.

You have error locator polynomial!

Where oh where have my packets gone wrong?

Factor? Sure.

Check all values?
Check your understanding.

You have error locator polynomial!
Where oh where have my packets gone wrong?
Factor? Sure.
Check all values? Sure.
Check your understanding.

You have error locator polynomial!
Where oh where have my packets gone wrong?
Factor? Sure.
Check all values? Sure.
Check your understanding.

You have error locator polynomial!
Where oh where have my packets gone wrong?
Factor? Sure.
Check all values? Sure.
Efficiency?
Check your understanding.

You have error locator polynomial!
Where oh where have my packets gone wrong?
Factor? Sure.
Check all values? Sure.
Efficiency? Sure.
Check your understanding.

You have error locator polynomial!
Where oh where have my packets gone wrong?
Factor? Sure.
Check all values? Sure.
Efficiency? Sure. Only $n + 2k$ values.
Check your understanding.

You have error locator polynomial!

Where oh where have my packets gone wrong?

Factor? Sure.
Check all values? Sure.

Efficiency? Sure. Only \( n + 2k \) values.
See where it is 0.
Hmmm...

Is there one and only one $P(x)$ from Berlekamp-Welsh procedure?
Hmmm...

Is there one and only one $P(x)$ from Berlekamp-Welsh procedure?

**Existence:** there is a $P(x)$ and $E(x)$ that satisfy equations.
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).$$  \hspace{1cm} (1)
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).$$

(1)

**Proof:**
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

\[
\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x). \quad (1)
\]

**Proof:**
We claim
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).$$  \hspace{1cm} (1)

**Proof:**

We claim

$$Q'(x)E(x) = Q(x)E'(x) \text{ on } n + 2k \text{ values of } x.$$ \hspace{1cm} (2)
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).$$

(1)

**Proof:**

We claim

$$Q'(x)E(x) = Q(x)E'(x)$$

on $n + 2k$ values of $x$.  

(2)

Equation 2 implies 1:
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).$$  \hspace{1cm} (1)

**Proof:**
We claim

$$Q'(x)E(x) = Q(x)E'(x) \text{ on } n+2k \text{ values of } x.$$  \hspace{1cm} (2)

Equation 2 implies 1:
$Q'(x)E(x)$ and $Q(x)E'(x)$ are degree $n+2k-1$. 

**Unique solution for** $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).$$

(1)

**Proof:**

We claim

$$Q'(x)E(x) = Q(x)E'(x) \text{ on } n+2k \text{ values of } x.$$  

(2)

Equation 2 implies 1:

$Q'(x)E(x)$ and $Q(x)E'(x)$ are degree $n+2k-1$

and agree on $n+2k$ points
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).$$  \hspace{1cm} (1)

**Proof:**
We claim

$$Q'(x)E(x) = Q(x)E'(x) \text{ on } n+2k \text{ values of } x.$$ \hspace{1cm} (2)

Equation 2 implies 1:

$Q'(x)E(x)$ and $Q(x)E'(x)$ are degree $n+2k-1$
and agree on $n+2k$ points

$E(x)$ and $E'(x)$ have at most $k$ zeros each.
Unique solution for \( P(x) \)

**Uniqueness:** any solution \( Q'(x) \) and \( E'(x) \) have

\[
\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).
\]  

(1)

**Proof:**

We claim

\[
Q'(x)E(x) = Q(x)E'(x) \text{ on } n+2k \text{ values of } x.
\]  

(2)

Equation 2 implies 1:

\( Q'(x)E(x) \) and \( Q(x)E'(x) \) are degree \( n+2k-1 \)

and agree on \( n+2k \) points

\( E(x) \) and \( E'(x) \) have at most \( k \) zeros each.

Can cross divide at \( n \) points.
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).$$ \hspace{1cm} (1)

**Proof:**

We claim

$$Q'(x)E(x) = Q(x)E'(x) \text{ on } n+2k \text{ values of } x.$$ \hspace{1cm} (2)

Equation 2 implies 1:

$Q'(x)E(x)$ and $Q(x)E'(x)$ are degree $n+2k-1$

and agree on $n+2k$ points

$E(x)$ and $E'(x)$ have at most $k$ zeros each.

Can cross divide at $n$ points.

$$\Rightarrow \frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} \text{ equal on } n \text{ points.}$$
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).$$

(1)

**Proof:**

We claim

$$Q'(x)E(x) = Q(x)E'(x)$$

on $n + 2k$ values of $x$. (2)

Equation 2 implies 1:

$Q'(x)E(x)$ and $Q(x)E'(x)$ are degree $n + 2k - 1$

and agree on $n + 2k$ points

$E(x)$ and $E'(x)$ have at most $k$ zeros each.

Can cross divide at $n$ points.

$$\Rightarrow \quad \frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)}$$

equal on $n$ points.

Both degree $\leq n$
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).$$

(1)

**Proof:**
We claim

$$Q'(x)E(x) = Q(x)E'(x)$$

on $n + 2k$ values of $x$. (2)

Equation 2 implies 1:

$Q'(x)E(x)$ and $Q(x)E'(x)$ are degree $n + 2k - 1$ and agree on $n + 2k$ points.

$E(x)$ and $E'(x)$ have at most $k$ zeros each. Can cross divide at $n$ points.

$\implies \frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)}$ equal on $n$ points.

Both degree $\leq n \implies$ Same polynomial!
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).$$  \hspace{1cm} (1)

**Proof:**

We claim

$$Q'(x)E(x) = Q(x)E'(x) \text{ on } n + 2k \text{ values of } x. \hspace{1cm} (2)$$

Equation 2 implies 1:

$Q'(x)E(x)$ and $Q(x)E'(x)$ are degree $n + 2k - 1$
and agree on $n + 2k$ points

$E(x)$ and $E'(x)$ have at most $k$ zeros each.

Can cross divide at $n$ points.

$$\implies \frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} \text{ equal on } n \text{ points.}$$

Both degree $\leq n \implies$ Same polynomial!
Fact: \( Q'(x)E(x) = Q(x)E'(x) \) on \( n + 2k \) values of \( x \).
Fact: $Q'(x)E(x) = Q(x)E'(x)$ on $n+2k$ values of $x$.

Proof:
Last bit.

Fact: $Q'(x)E(x) = Q(x)E'(x)$ on $n + 2k$ values of $x$.

Proof: Construction implies that
Fact: $Q'(x)E(x) = Q(x)E'(x)$ on $n + 2k$ values of $x$.

Proof: Construction implies that

$$Q(i) = R(i)E(i)$$

$$Q'(i) = R(i)E'(i)$$
Fact: $Q'(x)E(x) = Q(x)E'(x)$ on $n + 2k$ values of $x$.

Proof: Construction implies that

$$Q(i) = R(i)E(i)$$
$$Q'(i) = R(i)E'(i)$$

for $i \in \{1, \ldots, n + 2k\}$. 
Fact: $Q'(x)E(x) = Q(x)E'(x)$ on $n + 2k$ values of $x$.

Proof: Construction implies that

\[
Q(i) = R(i)E(i) \\
Q'(i) = R(i)E'(i)
\]

for $i \in \{1, \ldots n + 2k\}$.

If $E(i) = 0$, then $Q(i) = 0$. 
Fact: \( Q'(x)E(x) = Q(x)E'(x) \) on \( n + 2k \) values of \( x \).

Proof: Construction implies that

\[
Q(i) = R(i)E(i)
\]

\[
Q'(i) = R(i)E'(i)
\]

for \( i \in \{1, \ldots n + 2k\} \).

If \( E(i) = 0 \), then \( Q(i) = 0 \). If \( E'(i) = 0 \), then \( Q'(i) = 0 \).
Fact: $Q'(x)E(x) = Q(x)E'(x)$ on $n + 2k$ values of $x$.

Proof: Construction implies that

$$Q(i) = R(i)E(i)$$
$$Q'(i) = R(i)E'(i)$$

for $i \in \{1, \ldots n + 2k\}$.

If $E(i) = 0$, then $Q(i) = 0$. If $E'(i) = 0$, then $Q'(i) = 0$.

$$\implies Q(i)E'(i) = Q'(i)E(i)$$

holds when $E(i)$ or $E'(i)$ are zero.
Fact: $Q'(x)E(x) = Q(x)E'(x)$ on $n + 2k$ values of $x$.

Proof: Construction implies that

$$Q(i) = R(i)E(i)$$
$$Q'(i) = R(i)E'(i)$$

for $i \in \{1, \ldots, n+2k\}$.

If $E(i) = 0$, then $Q(i) = 0$. If $E'(i) = 0$, then $Q'(i) = 0$. 

$$\implies Q(i)E'(i) = Q'(i)E(i)$$ holds when $E(i)$ or $E'(i)$ are zero.

When $E'(i)$ and $E(i)$ are not zero
Fact: \( Q'(x)E(x) = Q(x)E'(x) \) on \( n + 2k \) values of \( x \).

Proof: Construction implies that

\[
Q(i) = R(i)E(i) \\
Q'(i) = R(i)E'(i)
\]

for \( i \in \{1, \ldots, n + 2k\} \).

If \( E(i) = 0 \), then \( Q(i) = 0 \). If \( E'(i) = 0 \), then \( Q'(i) = 0 \).

\[\implies Q(i)E'(i) = Q'(i)E(i) \] holds when \( E(i) \) or \( E'(i) \) are zero.

When \( E'(i) \) and \( E(i) \) are not zero

\[
\frac{Q'(i)}{E'(i)} = \frac{Q(i)}{E(i)} = R(i).
\]
Fact: $Q'(x)E(x) = Q(x)E'(x)$ on $n + 2k$ values of $x$.

Proof: Construction implies that

$$Q(i) = R(i)E(i)$$
$$Q'(i) = R(i)E'(i)$$

for $i \in \{1, \ldots n + 2k\}$.

If $E(i) = 0$, then $Q(i) = 0$. If $E'(i) = 0$, then $Q'(i) = 0$.

$$\implies Q(i)E'(i) = Q'(i)E(i)$$

holds when $E(i)$ or $E'(i)$ are zero.

When $E'(i)$ and $E(i)$ are not zero

$$\frac{Q'(i)}{E'(i)} = \frac{Q(i)}{E(i)} = R(i).$$

Cross multiplying gives equality in fact for these points.
Fact: \( Q'(x)E(x) = Q(x)E'(x) \) on \( n + 2k \) values of \( x \).

Proof: Construction implies that

\[
Q(i) = R(i)E(i) \\
Q'(i) = R(i)E'(i)
\]

for \( i \in \{1, \ldots n + 2k\} \).

If \( E(i) = 0 \), then \( Q(i) = 0 \). If \( E'(i) = 0 \), then \( Q'(i) = 0 \).

\[ \implies Q(i)E'(i) = Q'(i)E(i) \]

holds when \( E(i) \) or \( E'(i) \) are zero.

When \( E'(i) \) and \( E(i) \) are not zero

\[
\frac{Q'(i)}{E'(i)} = \frac{Q(i)}{E(i)} = R(i).
\]

Cross multiplying gives equality in fact for these points. \( \square \)
Fact: \( Q'(x)E(x) = Q(x)E'(x) \) on \( n + 2k \) values of \( x \).

Proof: Construction implies that

\[
Q(i) = R(i)E(i) \\
Q'(i) = R(i)E'(i)
\]

for \( i \in \{1, \ldots n + 2k\} \).

If \( E(i) = 0 \), then \( Q(i) = 0 \). If \( E'(i) = 0 \), then \( Q'(i) = 0 \).

\[
\implies Q(i)E'(i) = Q'(i)E(i) \text{ holds when } E(i) \text{ or } E'(i) \text{ are zero.}
\]

When \( E'(i) \) and \( E(i) \) are not zero

\[
\frac{Q'(i)}{E'(i)} = \frac{Q(i)}{E(i)} = R(i).
\]

Cross multiplying gives equality in fact for these points. \( \square \)

Points to polynomials, have to deal with zeros!
Fact: $Q'(x)E(x) = Q(x)E'(x)$ on $n + 2k$ values of $x$.

Proof: Construction implies that

$$Q(i) = R(i)E(i)$$
$$Q'(i) = R(i)E'(i)$$

for $i \in \{1, \ldots n + 2k\}$.

If $E(i) = 0$, then $Q(i) = 0$. If $E'(i) = 0$, then $Q'(i) = 0$.

$$\implies Q(i)E'(i) = Q'(i)E(i)$$

holds when $E(i)$ or $E'(i)$ are zero.

When $E'(i)$ and $E(i)$ are not zero

$$\frac{Q'(i)}{E'(i)} = \frac{Q(i)}{E(i)} = R(i).$$

Cross multiplying gives equality in fact for these points.

Points to polynomials, have to deal with zeros!

Example: dealing with $\frac{x-2}{x-2}$ at $x = 2$.  

□
Berlekamp-Welsh algorithm decodes correctly when \( k \) errors!
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets?
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode?
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$. 
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree?
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover?
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

How many packets?
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

How many packets? $n + 2k$
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

How many packets? $n + 2k$
Why?
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

How many packets? $n + 2k$
Why?
  $k$ changes to make diff. messages overlap
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

How many packets? $n + 2k$
Why?
  $k$ changes to make different messages overlap
How to encode?
Quick Check. Error Correction.

Communicate \( n \) packets, with \( k \) erasures.

- How many packets? \( n + k \)
- How to encode? With polynomial, \( P(x) \).
- Of degree? \( n - 1 \)
- Recover? Reconstruct \( P(x) \) with any \( n \) points!

Communicate \( n \) packets, with \( k \) errors.

- How many packets? \( n + 2k \)
- Why?
  - \( k \) changes to make diff. messages overlap
- How to encode? With polynomial, \( P(x) \).
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

How many packets? $n + 2k$
Why?
  $k$ changes to make diff. messages overlap
How to encode? With polynomial, $P(x)$. Of degree?
Quick Check. Error Correction.

Communicate \( n \) packets, with \( k \) erasures.

How many packets? \( n + k \)
How to encode? With polynomial, \( P(x) \).
Of degree? \( n - 1 \)
Recover? Reconstruct \( P(x) \) with any \( n \) points!

Communicate \( n \) packets, with \( k \) errors.

How many packets? \( n + 2k \)
Why?
\( k \) changes to make diff. messages overlap
How to encode? With polynomial, \( P(x) \). Of degree? \( n - 1 \).
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

- How many packets? $n + k$
- How to encode? With polynomial, $P(x)$.
- Of degree? $n - 1$
- Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

- How many packets? $n + 2k$
- Why?
  - $k$ changes to make diff. messages overlap
- Recover?
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

How many packets? $n + 2k$
Why?
   $k$ changes to make diff. messages overlap
How to encode? With polynomial, $P(x)$. Of degree? $n - 1$.
Recover?
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

How many packets? $n + 2k$
Why?
  $k$ changes to make diff. messages overlap
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$.
Recover?
Reconstruct error polynomial, $E(X)$, and $P(x)$!
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

How many packets? $n + 2k$
Why?
    $k$ changes to make diff. messages overlap
How to encode? With polynomial, $P(x)$. Of degree? $n - 1$.
Recover?
Reconstruct error polynomial, $E(X)$, and $P(x)$!
    Nonlinear equations.
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

How many packets? $n + 2k$
Why?
$k$ changes to make diff. messages overlap
How to encode? With polynomial, $P(x)$. Of degree? $n - 1$.
Recover?
Reconstruct error polynomial, $E(X)$, and $P(x)$!
Nonlinear equations.
Reconstruct $E(x)$ and $Q(x) = E(x)P(x)$. 
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

How many packets? $n + 2k$
Why?
  $k$ changes to make diff. messages overlap
How to encode? With polynomial, $P(x)$. Of degree? $n - 1$.
Recover?
  Reconstruct error polynomial, $E(X)$, and $P(x)$!
  Nonlinear equations.
Reconstruct $E(x)$ and $Q(x) = E(x)P(x)$. Linear Equations.
Quick Check. Error Correction.

Communicate \(n\) packets, with \(k\) erasures.

How many packets? \(n + k\)
How to encode? With polynomial, \(P(x)\).
Of degree? \(n - 1\)
Recover? Reconstruct \(P(x)\) with any \(n\) points!

Communicate \(n\) packets, with \(k\) errors.

How many packets? \(n + 2k\)
Why?
\(k\) changes to make diff. messages overlap
How to encode? With polynomial, \(P(x)\). Of degree? \(n - 1\).
Recover?
Reconstruct error polynomial, \(E(X)\), and \(P(x)\)!
Nonlinear equations.
Reconstruct \(E(x)\) and \(Q(x) = E(x)P(x)\). Linear Equations.
Polynomial division!
Quick Check. Error Correction.

Communicate \( n \) packets, with \( k \) erasures.

- How many packets? \( n + k \)
- How to encode? With polynomial, \( P(x) \).
- Of degree? \( n - 1 \)
- Recover? Reconstruct \( P(x) \) with any \( n \) points!

Communicate \( n \) packets, with \( k \) errors.

- How many packets? \( n + 2k \)
- Why?
  \( k \) changes to make diff. messages overlap
- How to encode? With polynomial, \( P(x) \). Of degree? \( n - 1 \).
- Recover?
  - Reconstruct error polynomial, \( E(X) \), and \( P(x) \)!
    - Nonlinear equations.
  - Reconstruct \( E(x) \) and \( Q(x) = E(x)P(x) \). Linear Equations.
  - Polynomial division! \( P(x) = Q(x)/E(x) \)!
Quick Check. Error Correction.

Communicate \(n\) packets, with \(k\) erasures.

How many packets? \(n + k\)
How to encode? With polynomial, \(P(x)\).
Of degree? \(n - 1\)
Recover? Reconstruct \(P(x)\) with any \(n\) points!

Communicate \(n\) packets, with \(k\) errors.

How many packets? \(n + 2k\)
Why?
\(k\) changes to make diff. messages overlap
How to encode? With polynomial, \(P(x)\). Of degree? \(n - 1\).
Recover?
Reconstruct error polynomial, \(E(X)\), and \(P(x)\)!
Nonlinear equations.
Reconstruct \(E(x)\) and \(Q(x) = E(x)P(x)\). Linear Equations.
Polynomial division! \(P(x) = Q(x)/E(x)\)!

Reed-Solomon codes.
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

How many packets? $n + 2k$
Why?
  $k$ changes to make diff. messages overlap
How to encode? With polynomial, $P(x)$. Of degree? $n - 1$.
Recover?
  Reconstruct error polynomial, $E(X)$, and $P(x)$!
    Nonlinear equations.
Reconstruct $E(x)$ and $Q(x) = E(x)P(x)$. Linear Equations.
  Polynomial division! $P(x) = Q(x)/E(x)$!

Reed-Solomon codes. Welsh-Berlekamp Decoding.
Quick Check. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

How many packets? $n + 2k$
Why?
  $k$ changes to make diff. messages overlap
How to encode? With polynomial, $P(x)$. Of degree? $n - 1$.
Recover?
  Reconstrcut error polynomial, $E(X)$, and $P(x)$!
  Nonlinear equations.
Reconstruct $E(x)$ and $Q(x) = E(x)P(x)$. Linear Equations.
Polynomial division! $P(x) = Q(x)/E(x)$!

Reed-Solomon codes. Welsh-Berlekamp Decoding. Perfection!
Problem: Communicate $n$ packets $m_1, \ldots, m_n$ on noisy channel that corrupts $\leq k$ packets.
Problem: Communicate $n$ packets $m_1, \ldots, m_n$ on noisy channel that corrupts $\leq k$ packets.

Reed-Solomon Code:
Reed-Solomon code.

**Problem:** Communicate $n$ packets $m_1, \ldots, m_n$ on noisy channel that corrupts $\leq k$ packets.

**Reed-Solomon Code:**

1. Make a polynomial, $P(x)$ of degree $n - 1$, that encodes message: coefficients, $p_0, \ldots, p_{n-1}$. 

Reed-Solomon code.

**Problem:** Communicate $n$ packets $m_1, \ldots, m_n$ on noisy channel that corrupts $\leq k$ packets.

**Reed-Solomon Code:**

1. Make a polynomial, $P(x)$ of degree $n - 1$, that encodes message: coefficients, $p_0, \ldots, p_{n-1}$.
2. Send $P(1), \ldots, P(n + 2k)$. 