Today.

Last time:
Today.

Last time:
Shared (and sort of kept) secrets.
Today.

Last time:
Shared (and sort of kept) secrets.
Today.

Last time:
  Shared (and sort of kept) secrets.

Today: Errors
Today.

Last time:
  Shared (and sort of kept) secrets.

Today: Errors
  Tolerate Loss: erasure codes.
Today.

Last time:
  Shared (and sort of kept) secrets.

Today: Errors
  Tolerate Loss: erasure codes.
  Tolerate corruption!
Today.

Last time:
  Shared (and sort of kept) secrets.

Today: Errors
  Tolerate Loss: erasure codes.
  Tolerate corruption!
Poll

**Line:** $y = mx + b$
Poll

Line: \( y = mx + b \)

Poly: \( y = P(x) = a_d x^d + a_{d-1} x^{d-1} \ldots a_0 x^0 \)
Poll

Line: \( y = mx + b \)

Poly: "\( y \)" = \( P(x) = a_dx^d + a_{d-1}x^{d-1} \ldots a_0x^0 \)

Everything below is true. Mark if you know it and perhaps why it is true.

(A) Two points determine a line: \( mx + b \)
(B) A root of \( P(x) \), is a where \( P(a) = 0 \).
(C) A degree \( d \) polynomial has at most \( d \) roots.
(D) Arithmetic modulo a prime \( p \) is a "field".
Poll

Line: \( y = mx + b \)

Poly: "y" = \( P(x) = a_d x^d + a_{d-1} x^{d-1} ... a_0 x^0 \)

Everything below is true. Mark if you know it and perhaps why it is true.

(A) Two points determine a line: \( mx + b \)
(B) A root of \( P(x) \), is a where \( P(a) = 0 \).
(C) A degree d polynomial has at most d roots.
(D) Arithmetic modulo a prime p is a "field".

(A) If a polynomial has a root at \( a \), \( P(x) = Q(x)(x - a) \).
(B) A line has at most one root, if not always zero.
(C) System: \( y_1 = mx_1 + b, y_2 = mx_2 + b \) has unique solution \((m, b)\).
(D) Degree of a polynomial \( P(x)^2 \) is \( 2d \) if \( P(x) \) is degree \( d \).
Line: \( y = mx + b \)

Poly: \( "y" = P(x) = a_d x^d + a_{d-1} x^{d-1} \ldots a_0 x^0 \)

Everything below is true. Mark if you know it and perhaps why it is true.

(A) Two points determine a line: \( mx + b \)

(B) A root of \( P(x) \), is a where \( P(a) = 0 \).

(C) A degree d polynomial has at most d roots.

(D) Arithmetic modulo a prime p is a "field".

(A) If a polynomial has a root at \( a \), \( P(x) = Q(x)(x - a) \).

(B) A line has at most one root, if not always zero.

(C) System: \( y_1 = mx_1 + b, y_2 = mx_2 + b \) has unique solution \((m, b)\).

(D) Degree of a polynomial \( P(x)^2 \) is \( 2d \) if \( P(x) \) is degree \( d \).

(C) may not be true.
The mathematics.

There is a unique polynomial of degree $d$ that contains any $d + 1$ points.
The mathematics.

There is a unique polynomial of degree $d$ that contains any $d + 1$ points.

Assumption: a field, in particular, arithmetic mod $p$. 
The mathematics.

There is a unique polynomial of degree $d$ that contains any $d + 1$ points.

Assumption: a field, in particular, arithmetic $\mod p$.

Big Idea:
The mathematics.

There is a unique polynomial of degree $d$ that contains any $d + 1$ points.

Assumption: a field, in particular, arithmetic $\mod p$.

Big Idea:

A polynomial: $P(x) = a_d x^d + \cdots a_0$ has $d + 1$ coefficients.
The mathematics.

There is a unique polynomial of degree $d$ that contains any $d + 1$ points.

Assumption: a field, in particular, arithmetic $\mod p$.

Big Idea:

A polynomial: $P(x) = a_dx^d + \cdots + a_0$ has $d + 1$ coefficients. Any set of $d + 1$ points determines the polynomial.
The mathematics.

There is a unique polynomial of degree $d$ that contains any $d + 1$ points.

Assumption: a field, in particular, arithmetic $\mod p$.

Big Idea:

A polynomial: $P(x) = a_d x^d + \cdots a_0$ has $d + 1$ coefficients. Any set of $d + 1$ points determines the polynomial.
The mathematics.

There is a unique polynomial of degree $d$ that contains any $d + 1$ points.

Assumption: a field, in particular, arithmetic $\mod p$.

Big Idea:

A polynomial: $P(x) = a_dx^d + \cdots a_0$ has $d + 1$ coefficients.
Any set of $d + 1$ points determines the polynomial.

Stare at the above.
The mathematics.

There is a unique polynomial of degree $d$ that contains any $d + 1$ points.

Assumption: a field, in particular, arithmetic mod $p$.

Big Idea:

A polynomial: $P(x) = a_d x^d + \cdots a_0$ has $d + 1$ coefficients.

Any set of $d + 1$ points determines the polynomial.

Stare at the above. What does it mean?
The mathematics.

There is a unique polynomial of degree $d$ that contains any $d + 1$ points.

Assumption: a field, in particular, arithmetic $\mod p$.

Big Idea:

A polynomial: $P(x) = a_d x^d + \cdots + a_0$ has $d + 1$ coefficients.

Any set of $d + 1$ points determines the polynomial.

Stare at the above. What does it mean?
Many representations of a polynomial!
The mathematics.

There is a unique polynomial of degree $d$ that contains any $d+1$ points.

Assumption: a field, in particular, arithmetic $\mod p$.

Big Idea:

A polynomial: $P(x) = a_d x^d + \cdots + a_0$ has $d+1$ coefficients. Any set of $d+1$ points determines the polynomial.

Stare at the above. What does it mean?
Many representations of a polynomial!
One coefficient representation.
The mathematics.

There is a unique polynomial of degree $d$ that contains any $d + 1$ points.

Assumption: a field, in particular, arithmetic $\text{mod } p$.

Big Idea:

A polynomial: $P(x) = a_d x^d + \cdots + a_0$ has $d + 1$ coefficients. Any set of $d + 1$ points determines the polynomial.

Stare at the above. What does it mean?
Many representations of a polynomial!
One coefficient representation.
Many, many point,value representations.
The mathematics.

There is a unique polynomial of degree $d$ that contains any $d+1$ points.

Assumption: a field, in particular, arithmetic $\mod p$.

Big Idea:

A polynomial: $P(x) = a_d x^d + \cdots a_0$ has $d + 1$ coefficients.

Any set of $d + 1$ points determines the polynomial.

Stare at the above. What does it mean?

Many representations of a polynomial!

One coefficient representation.

Many, many point,value representations.
The mathematics.

**There is a unique polynomial of degree** $d$ **that contains any** $d + 1$ **points.**

Assumption: a field, in particular, arithmetic $\mod p$.

Big Idea:

A polynomial: $P(x) = a_d x^d + \cdots a_0$ has $d + 1$ coefficients. 
**Any set of** $d + 1$ **points** determines the polynomial.

Stare at the above. What does it mean? 
Many representations of a polynomial! 
One coefficient representation. 
Many, many point,value representations.

Some details:
The mathematics.

There is a unique polynomial of degree $d$ that contains any $d + 1$ points.

Assumption: a field, in particular, arithmetic $\mod p$.

Big Idea:

A polynomial: $P(x) = a_dx^d + \cdots a_0$ has $d + 1$ coefficients. Any set of $d + 1$ points determines the polynomial.

Stare at the above. What does it mean?

Many representations of a polynomial!

One coefficient representation.

Many, many point,value representations.

Some details:

Degree $d$ generally degree “at most” $d$. 
The mathematics.

There is a unique polynomial of degree $d$ that contains any $d+1$ points.

Assumption: a field, in particular, arithmetic mod $p$.

Big Idea:

A polynomial: $P(x) = a_dx^d + \cdots + a_0$ has $d+1$ coefficients. Any set of $d+1$ points determines the polynomial.

Stare at the above. What does it mean?

Many representations of a polynomial!

One coefficient representation.

Many, many point, value representations.

Some details:

Degree $d$ generally degree “at most” $d$.

(example: choose 10 points on a line.)
The mathematics.

There is a unique polynomial of degree $d$ that contains any $d+1$ points.

Assumption: a field, in particular, arithmetic $\mod p$.

Big Idea:

A polynomial: $P(x) = a_dx^d + \cdots a_0$ has $d+1$ coefficients.

Any set of $d+1$ points determines the polynomial.

Stare at the above. What does it mean?

Many representations of a polynomial!

One coefficient representation.

Many, many point, value representations.

Some details:

Degree $d$ generally degree “at most” $d$.

(example: choose 10 points on a line.)

Arithmetic $(\mod p) \implies$ work with $O(\log p)$ bit numbers.
Proof sketches.

**Property 2** A polynomial: \( P(x) = a_d x^d + \cdots a_0 \) has \( d + 1 \) coefficients.
Proof sketches.

**Property 2** A polynomial: \( P(x) = a_d x^d + \cdots a_0 \) has \( d + 1 \) coefficients. Any set of \( d + 1 \) points uniquely determines the polynomial.
Proof sketches.

Property 2 A polynomial: \( P(x) = a_d x^d + \cdots a_0 \) has \( d + 1 \) coefficients. Any set of \( d + 1 \) points uniquely determines the polynomial.
Proof sketches.

**Property 2** A polynomial: \( P(x) = a_dx^d + \cdots + a_0 \) has \( d+1 \) coefficients. Any set of \( d+1 \) points uniquely determines the polynomial.

Existence: Lagrange Interpolation.
Proof sketches.

**Property 2** A polynomial: \( P(x) = a_d x^d + \cdots a_0 \) has \( d + 1 \) coefficients. Any set of \( d + 1 \) points uniquely determines the polynomial.

Existence: Lagrange Intropolation.
Degree \( d \), \( \Delta_i(x) \) polynomials.
Proof sketches.

**Property 2** A polynomial: \( P(x) = a_d x^d + \cdots a_0 \) has \( d + 1 \) coefficients. Any set of \( d + 1 \) points uniquely determines the polynomial.

Existence: Lagrange Interpolation.
Degree \( d \), \( \Delta_i(x) \) polynomials.
Factors of \((x - x_j)\) to zero out at \( x_j \neq x_i \).
Proof sketches.

**Property 2** A polynomial: \( P(x) = a_dx^d + \cdots + a_0 \) has \( d + 1 \) coefficients. Any set of \( d + 1 \) points uniquely determines the polynomial.

Existence: Lagrange Interpolation.
Degree \( d \), \( \Delta_i(x) \) polynomials.
Factors of \( (x - x_j) \) to zero out at \( x_j \neq x_i \).
Multiply by zero. My love is won.
Combine.
Proof sketches.

**Property 2** A polynomial: \( P(x) = a_d x^d + \cdots a_0 \) has \( d + 1 \) coefficients. Any set of \( d + 1 \) points uniquely determines the polynomial.

Existence: Lagrange Interpolation.

Degree \( d \), \( \Delta_i(x) \) polynomials.

factors of \((x - x_j)\) to zero out at \( x_j \neq x_i \).

Multiply by zero. My love is won.

Combine.

Uniqueness:
Proof sketches.

Property 2 A polynomial: $P(x) = a_d x^d + \cdots + a_0$ has $d + 1$ coefficients. Any set of $d + 1$ points uniquely determines the polynomial.

Existence: Lagrange Interpolation.
Degree $d$, $\Delta_i(x)$ polynomials.
  factors of $(x - x_j)$ to zero out at $x_j \neq x_i$.
  Multiply by zero. My love is won.

Combine.

Uniqueness:
  Property 1 A non-zero degree $d$ polynomial has at most $d$ roots.
Proof sketches.

**Property 2** A polynomial: \( P(x) = a_dx^d + \cdots a_0 \) has \( d + 1 \) coefficients. Any set of \( d + 1 \) points uniquely determines the polynomial.

Existence: Lagrange Intropolation.
Degree \( d \), \( \Delta_i(x) \) polynomials.
  factors of \((x - x_j)\) to zero out at \( x_j \neq x_i \).
  Multiply by zero. My love is won.
Combine.

Uniqueness:
**Property 1** A non-zero degree \( d \) polynomial has at most \( d \) roots.
Factoring: \( P(x) \) with roots \( r_1, \ldots, r_d \)
Proof sketches.

**Property 2** A polynomial: \( P(x) = a_d x^d + \cdots a_0 \) has \( d + 1 \) coefficients. Any set of \( d + 1 \) points uniquely determines the polynomial.

Existence: Lagrange Intropolation.
- Degree \( d \), \( \Delta_i(x) \) polynomials.
  - factors of \( (x - x_j) \) to zero out at \( x_j \neq x_i \).
  
  Multiply by zero. My love is won.

Combine.

Uniqueness:
**Property 1** A non-zero degree \( d \) polynomial has at most \( d \) roots.
- Factoring: \( P(x) \) with roots \( r_1, \ldots, r_d \)
  
  \[ \implies P(x) = c(x - r_0)(x - r_1)\cdots(x - r_d). \]
Proof sketches.

**Property 2** A polynomial: \( P(x) = a_d x^d + \cdots a_0 \) has \( d + 1 \) coefficients. Any set of \( d + 1 \) points uniquely determines the polynomial.

Existence: Lagrange Interpolation.
Degree \( d \), \( \Delta_i(x) \) polynomials.
- factors of \((x - x_j)\) to zero out at \( x_j \neq x_i \).
- Multiply by zero. My love is won.

Combine.

Uniqueness:

**Property 1** A non-zero degree \( d \) polynomial has at most \( d \) roots.
Factoring: \( P(x) \) with roots \( r_1, \ldots, r_d \)
- \( \implies P(x) = c(x - r_0)(x - r_1) \cdots (x - r_d) \).

Love me some contradiction!
Proof sketches.

**Property 2** A polynomial: \( P(x) = a_d x^d + \cdots a_0 \) has \( d + 1 \) coefficients. Any set of \( d + 1 \) points uniquely determines the polynomial.

Existence: Lagrange Interpolation.
- Degree \( d \), \( \Delta_i(x) \) polynomials.
  - factors of \( (x - x_j) \) to zero out at \( x_j \neq x_i \).
  - Multiply by zero. My love is won.
- Combine.

Uniqueness:
- **Property 1** A non-zero degree \( d \) polynomial has at most \( d \) roots.
  - Factoring: \( P(x) \) with roots \( r_1, \ldots, r_d \)
  \[ \implies P(x) = c(x - r_0)(x - r_1)\cdots(x - r_d). \]
  - Love me some contradiction!
  - Two polynomials: \( P(x), Q(x) \), \( P(x) - Q(x) \) has too many roots.
Finite Fields

Proof works for reals, rationals, and complex numbers.
Finite Fields

Proof works for reals, rationals, and complex numbers.
..but not for integers, since no multiplicative inverses.
Proof works for reals, rationals, and complex numbers. 
..but not for integers, since no multiplicative inverses.
Arithmetic modulo a prime $p$ has multiplicative inverses.
Proof works for reals, rationals, and complex numbers. 
..but not for integers, since no multiplicative inverses. 
Arithmetic modulo a prime $p$ has multiplicative inverses.. 
..and has only a finite number of elements.
Finite Fields

Proof works for reals, rationals, and complex numbers.
..but not for integers, since no multiplicative inverses.
Arithmetic modulo a prime $p$ has multiplicative inverses..
..and has only a finite number of elements.
Good for computer science.
Finite Fields

Proof works for reals, rationals, and complex numbers.
..but not for integers, since no multiplicative inverses.
Arithmetic modulo a prime $p$ has multiplicative inverses..
..and has only a finite number of elements.
Good for computer science.
Arithmetic modulo a prime $m$ is a finite field denoted by $F_m$ or $GF(m)$. 
Proof works for reals, rationals, and complex numbers.  
..but not for integers, since no multiplicative inverses.  
Arithmetic modulo a prime $p$ has multiplicative inverses..  
..and has only a finite number of elements.  
Good for computer science.  
Arithmetic modulo a prime $m$ is a **finite field** denoted by $F_m$ or $GF(m)$.  
Intuitively, a field is a set with operations corresponding to addition, multiplication, and division.
Secret Sharing

**Modular Arithmetic Fact:** Exactly one polynomial degree $\leq d$ over $GF(p)$, $P(x)$, that hits $d + 1$ points.
Secret Sharing

**Modular Arithmetic Fact:** Exactly one polynomial degree $\leq d$ over $GF(p)$, $P(x)$, that hits $d+1$ points.

**Shamir’s $k$ out of $n$ Scheme:**
Secret Sharing

**Modular Arithmetic Fact:** Exactly one polynomial degree \( \leq d \) over \( GF(p) \), \( P(x) \), that hits \( d + 1 \) points.

**Shamir’s \( k \) out of \( n \) Scheme:**
Secret \( s \in \{0, \ldots, p - 1\} \)
Secret Sharing

**Modular Arithmetic Fact:** Exactly one polynomial degree \( \leq d \) over \( GF(p) \), \( P(x) \), that hits \( d + 1 \) points.

**Shamir’s \( k \) out of \( n \) Scheme:**
Secret \( s \in \{0, \ldots, p-1\} \)

1. Choose \( a_0 = s \), and randomly \( a_1, \ldots, a_{k-1} \).
Secret Sharing

**Modular Arithmetic Fact:** Exactly one polynomial degree \( \leq d \) over \( GF(p) \), \( P(x) \), that hits \( d + 1 \) points.

**Shamir’s \( k \) out of \( n \) Scheme:**
Secret \( s \in \{0, \ldots, p-1\} \)

1. Choose \( a_0 = s \), and randomly \( a_1, \ldots, a_{k-1} \).
2. Let \( P(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + \cdots a_0 \) with \( a_0 = s \).
Secret Sharing

Modular Arithmetic Fact: Exactly one polynomial degree $\leq d$ over $GF(p)$, $P(x)$, that hits $d+1$ points.

Shamir’s $k$ out of $n$ Scheme:
Secret $s \in \{0, \ldots, p-1\}$

1. Choose $a_0 = s$, and randomly $a_1, \ldots, a_{k-1}$.
2. Let $P(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + \cdots + a_0$ with $a_0 = s$.
3. Share $i$ is point $(i, P(i) \mod p)$.
Secret Sharing

**Modular Arithmetic Fact:** Exactly one polynomial degree \( \leq d \) over \( GF(p) \), \( P(x) \), that hits \( d + 1 \) points.

**Shamir’s k out of n Scheme:**

Secret \( s \in \{0, \ldots, p - 1\} \)

1. Choose \( a_0 = s \), and randomly \( a_1, \ldots, a_{k-1} \).
2. Let \( P(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + \cdots + a_0 \) with \( a_0 = s \).
3. Share \( i \) is point \((i, P(i) \mod p)\).
Secret Sharing

**Modular Arithmetic Fact:** Exactly one polynomial degree $\leq d$ over $GF(p)$, $P(x)$, that hits $d + 1$ points.

**Shamir’s $k$ out of $n$ Scheme:**
Secret $s \in \{0, \ldots, p-1\}$

1. Choose $a_0 = s$, and randomly $a_1, \ldots, a_{k-1}$.
2. Let $P(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + \cdots a_0$ with $a_0 = s$.
3. Share $i$ is point $(i, P(i) \mod p)$.

**Robustness:** Any $k$ knows secret.
Secret Sharing

**Modular Arithmetic Fact:** Exactly one polynomial degree \( \leq d \) over \( GF(p) \), \( P(x) \), that hits \( d + 1 \) points.

**Shamir’s \( k \) out of \( n \) Scheme:**
Secret \( s \in \{0, \ldots, p - 1\} \)

1. Choose \( a_0 = s \), and randomly \( a_1, \ldots, a_{k-1} \).
2. Let \( P(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + \cdots + a_0 \) with \( a_0 = s \).
3. Share \( i \) is point \((i, P(i) \mod p)\).

**Roubustness:** Any \( k \) knows secret.
Knowing \( k \) pts, only one \( P(x) \), evaluate \( P(0) \).

**Secrecy:** Any \( k - 1 \) knows nothing.
Secret Sharing

**Modular Arithmetic Fact:** Exactly one polynomial degree \( \leq d \) over \( GF(p) \), \( P(x) \), that hits \( d + 1 \) points.

**Shamir’s \( k \) out of \( n \) Scheme:**
Secret \( s \in \{0, \ldots, p - 1\} \)

1. Choose \( a_0 = s \), and randomly \( a_1, \ldots, a_{k-1} \).
2. Let \( P(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + \cdots a_0 \) with \( a_0 = s \).
3. Share \( i \) is point \((i, P(i) \mod p)\).

**Roubustness:** Any \( k \) knows secret.
Knowing \( k \) pts, only one \( P(x) \), evaluate \( P(0) \).

**Secrecy:** Any \( k - 1 \) knows nothing.
Knowing \( \leq k - 1 \) pts, any \( P(0) \) is possible.
Secret Sharing

**Modular Arithmetic Fact:** Exactly one polynomial degree $\leq d$ over $GF(p)$, $P(x)$, that hits $d + 1$ points.

**Shamir’s $k$ out of $n$ Scheme:**
Secret $s \in \{0, \ldots, p - 1\}$

1. Choose $a_0 = s$, and randomly $a_1, \ldots, a_{k-1}$.
2. Let $P(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + \cdots + a_0$ with $a_0 = s$.
3. Share $i$ is point $(i, P(i) \mod p)$.

**Robustness:** Any $k$ knows secret.
Knowing $k$ pts, only one $P(x)$, evaluate $P(0)$.

**Secrecy:** Any $k - 1$ knows nothing.
Knowing $\leq k - 1$ pts, any $P(0)$ is possible.

Two points make a line: the value of one point allows any y-intercept.
Secret Sharing

**Modular Arithmetic Fact:** Exactly one polynomial degree $\leq d$ over $GF(p)$, $P(x)$, that hits $d + 1$ points.

**Shamir’s $k$ out of $n$ Scheme:**
Secret $s \in \{0, \ldots, p-1\}$

1. Choose $a_0 = s$, and randomly $a_1, \ldots, a_{k-1}$.
2. Let $P(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + \cdots + a_0$ with $a_0 = s$.
3. Share $i$ is point $(i, P(i) \mod p)$.

**Robustness:** Any $k$ knows secret.
Knowing $k$ pts, only one $P(x)$, evaluate $P(0)$.

**Secrecy:** Any $k - 1$ knows nothing.
Knowing $\leq k - 1$ pts, any $P(0)$ is possible.

Two points make a line: the value of one point allows any y-intercept.

3 kids hand out 3 points. Any two know the line.
Minimality.

Need $p > n$ to hand out $n$ shares: $P(1) \ldots P(n)$. 

Minimality.

Need $p > n$ to hand out $n$ shares: $P(1) \ldots P(n)$.

For $b$-bit secret, must choose a prime $p > 2^b$. 
Minimality.

Need $p > n$ to hand out $n$ shares: $P(1) \ldots P(n)$.
For $b$-bit secret, must choose a prime $p > 2^b$.

**Theorem:** There is always a prime between $n$ and $2n$.

*Chebyshev said it,*

---

Chebyshev said it,
Minimality.

Need $p > n$ to hand out $n$ shares: $P(1) \ldots P(n)$.

For $b$-bit secret, must choose a prime $p > 2^b$.

**Theorem:** There is always a prime between $n$ and $2n$.

*Chebyshev said it,*

*And I say it again,*

"There is always a prime between $n$ and $2n."
Minimality.

Need $p > n$ to hand out $n$ shares: $P(1) \ldots P(n)$.

For $b$-bit secret, must choose a prime $p > 2^b$.

**Theorem:** There is always a prime between $n$ and $2n$.

*Chebyshev said it,*

*And I say it again,*

*There is always a prime*
Minimality.

Need $p > n$ to hand out $n$ shares: $P(1) \ldots P(n)$.

For $b$-bit secret, must choose a prime $p > 2^b$.

**Theorem:** There is always a prime between $n$ and $2n$.

*Chebyshev said it,*

*And I say it again,*

*There is always a prime*

*Between $n$ and $2n$.*
Minimality.

Need $p > n$ to hand out $n$ shares: $P(1) \ldots P(n)$.
For $b$-bit secret, must choose a prime $p > 2^b$.

**Theorem:** There is always a prime between $n$ and $2n$.
   *Chebyshev said it,*
   *And I say it again,*
   *There is always a prime*
   *Between $n$ and $2n.*

Working over numbers within 1 bit of secret size. **Minimality.**
Minimality.

Need \( p > n \) to hand out \( n \) shares: \( P(1) \ldots P(n) \).

For \( b \)-bit secret, must choose a prime \( p > 2^b \).

**Theorem:** There is always a prime between \( n \) and \( 2n \).

*Chebyshev said it,*
*And I say it again,*
*There is always a prime*
*Between \( n \) and \( 2n \).*

Working over numbers within 1 bit of secret size. **Minimality.**

With \( k \) shares, reconstruct polynomial, \( P(x) \).
Minimality.

Need $p > n$ to hand out $n$ shares: $P(1) \ldots P(n)$.

For $b$-bit secret, must choose a prime $p > 2^b$.

**Theorem:** There is always a prime between $n$ and $2n$.

*Chebyshev said it,*
*And I say it again,*
*There is always a prime*
*Between $n$ and $2n$.*

Working over numbers within 1 bit of secret size. **Minimality.**

With $k$ shares, reconstruct polynomial, $P(x)$.

With $k – 1$ shares, any of $p$ values possible for $P(0)$!
Minimality.

Need \( p > n \) to hand out \( n \) shares: \( P(1) \ldots P(n) \).
For \( b \)-bit secret, must choose a prime \( p > 2^b \).

**Theorem:** There is always a prime between \( n \) and \( 2n \).

*Chebyshev said it,*
*And I say it again,*
*There is always a prime*
*Between \( n \) and \( 2n \).*

Working over numbers within 1 bit of secret size. **Minimality.**

With \( k \) shares, reconstruct polynomial, \( P(x) \).
With \( k - 1 \) shares, any of \( p \) values possible for \( P(0) \)!
(Almost) any \( b \)-bit string possible!
Minimality.

Need \( p > n \) to hand out \( n \) shares: \( P(1) \ldots P(n) \).

For \( b \)-bit secret, must choose a prime \( p > 2^b \).

**Theorem:** There is always a prime between \( n \) and \( 2n \).

*Chebyshev said it,*
*And I say it again,*
*There is always a prime*
*Between \( n \) and \( 2n \).*

Working over numbers within 1 bit of secret size. **Minimality.**

With \( k \) shares, reconstruct polynomial, \( P(x) \).

With \( k - 1 \) shares, any of \( p \) values possible for \( P(0) \)!

(Almost) any \( b \)-bit string possible!

(Almost) the same as what is missing: one \( P(i) \).
Runtime.

1. Evaluate degree \( k - 1 \) polynomial \( n \) times using \( \log p \)-bit numbers.

2. Reconstruct secret by solving system of \( k \) equations using \( \log p \)-bit arithmetic.
Runtime: polynomial in \( k, n, \) and \( \log p \).

1. Evaluate degree \( k - 1 \) polynomial \( n \) times using \( \log p \)-bit numbers.

2. Reconstruct secret by solving system of \( k \) equations using \( \log p \)-bit arithmetic.
A bit more counting.

What is the number of degree $d$ polynomials over $GF(m)$?
A bit more counting.

What is the number of degree $d$ polynomials over $GF(m)$?

$\quad m^{d+1} \colon d + 1$ coefficients from $\{0, \ldots, m-1\}$. 
A bit more counting.

What is the number of degree $d$ polynomials over $GF(m)$?

- $m^{d+1}$: $d + 1$ coefficients from \{0, \ldots, m - 1\}.
- $m^{d+1}$: $d + 1$ points with $y$-values from \{0, \ldots, m - 1\}.
A bit more counting.

What is the number of degree $d$ polynomials over $GF(m)$?

- $m^{d+1}$: $d + 1$ coefficients from $\{0, \ldots, m-1\}$.
- $m^{d+1}$: $d + 1$ points with $y$-values from $\{0, \ldots, m-1\}$

Infinite number for reals, rationals, complex numbers!
Secret Sharing.

$n$ people, $k$ is enough.

(A) The modulus needs to be at least $n+1$.
(B) The modulus needs to be at least $k$.
(C) Use degree $k$ polynomial, hand out $n$ points.
(D) Use degree $n$ polynomial, hand out $k$ points.
(E) Use degree $k-1$ polynomial, hand out $n$ points.
(F) The modulus needs to be at least $2^s$, where $s$ is value of secret.
(G) The modulus needs to be at least $2^s$, where $s$ is size of secret.
Secret Sharing.

$n$ people, $k$ is enough.

(A) The modulus needs to be at least $n + 1$.
(B) The modulus needs to be at least $k$.
(C) Use degree $k$ polynomial, hand out $n$ points.
(D) Use degree $n$ polynomial, hand out $k$ points.
(E) Use degree $k - 1$ polynomial, hand out $n$ points.
(F) The modulus needs to be at least $2^s$, where $s$ is value of secret.
(G) The modulus needs to be at least $2^s$, where $s$ is size of secret.

(A), (B), (E), (F)
Erasure Codes.

Satellite

GPS device
Erasure Codes.

Satellite

3 packet message.

GPS device
Erasure Codes.

Satellite

GPS device

3 packet message.

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.

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.

\[ n \text{ packet message, channel that loses } k \text{ packets.} \]
Solution Idea.

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

$n$ 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.

$n$ 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!!!!!!
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.
The Scheme

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

Channel: Lossy channel: loses $k$ packets.
The Scheme

**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?
The Scheme

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!
The Scheme

**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 = \( m_0, m_1 \ldots, m_{n-1} \).

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 \( P(1), \ldots, P(n+k) \).
**The Scheme**

**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 = $m_0, m_1, \ldots, m_{n-1}$.

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

Any $n$ of the $n + k$ packets gives polynomial ...
The Scheme

**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 = \( m_0, m_1 \ldots, m_{n-1} \).

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

$G = \binom{n+k}{k}$

$\text{Satellite}$

$n$ packet message.

$\text{GPS device}$

Any $n$ packets is enough!
Erasure Codes.

Satellite

$n$ packet message.

Lose $k$ packets.

GPS device
Erasure Codes.

Satellite

1 2 \cdots n+k

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

Lose $k$ packets.

GPS device
Erasure Codes.

Satellite

1 2 ... n+k

GPS device

Lose $k$ packets.

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

  Lose k packets.

  Any n packets is enough!

- GPS device

  n packet message. So send n+k!
Erasure Codes.

Satellite

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

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

GPS device

\[ n \text{ packet message. So send } n+k! \]

Lose \( k \) packets.

Any \( n \) packets is enough!

\[ n \text{ packet message.} \]
Erasure Codes.

Satellite

1 2 \rightarrow \cdots \rightarrow n+k

GPS device

\begin{align*}
\text{Lose } k \text{ packets.} \\
\text{Any } n \text{ packets is enough!}
\end{align*}

n \text{ packet message. So send } n+k!

n \text{ packet message.}

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.
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.
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.
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.
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.
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.
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.
Erasure Code: Example.

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

Better work modulo 7 at least!

Why?

$(0, P(0)) = (5, P(5)) \pmod{5}$
Send message of 1, 4, and 4.
Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$.
How?
   Lagrange Interpolation.
   Linear System.

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.
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.

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

$P(x) = x^2 \pmod{5}$
$P(1) = 1$,
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.

\[ P(x) = x^2 \pmod{5} \]
\[ P(1) = 1, P(2) = 4, \]
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.

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

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

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.

$P(x) = x^2 \pmod{5}$
$P(1) = 1, P(2) = 4, P(3) = 9 = 4 \pmod{5}$
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 $(0, P(0)) \ldots (5, P(5))$. 
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.
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?
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? $(0, P(0)) = (5, P(5)) \pmod{5}$
Example

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

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.
Example

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

Linear equations:

$P(x) = a_2x^2 + a_1x + a_0 \equiv 1 \pmod{7}$

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

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

$6a_1 + 3a_0 \equiv 2 \pmod{7}$

$5a_1 + 4a_0 \equiv 0 \pmod{7}$

$a_1 \equiv 2 \pmod{7}$

$a_0 \equiv 2 \pmod{7}$

$a_2 \equiv 2 \pmod{7}$

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:

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

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

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

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

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

$6a_1 + 3a_0 \equiv 2 \pmod{7}$,
Example

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

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

$6a_1 + 3a_0 = 2 \pmod{7}, \quad 5a_1 + 4a_0 = 0 \pmod{7}$
Make polynomial with $P(1) = 1$, $P(2) = 4$, $P(3) = 4$. Modulo 7 to accommodate at least 6 packets.

Linear equations:

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

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

$a_1 = 2a_0$. 

Example

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

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

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

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

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

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

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

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

Linear equations:

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

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

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

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

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

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

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

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

Modulo 7 to accommodate at least 6 packets.

Linear equations:

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

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

\[
\begin{align*}
a_1 &= 2a_0. \\
a_0 &= 2 \pmod{7} \\
a_1 &= 4 \pmod{7} \\
a_2 &= 2 \pmod{7}
\end{align*}
\]

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

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

Linear equations:

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

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

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

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

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

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

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

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

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

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

\[
P(1) = 1, \quad P(2) = 4,
\]
Example

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

Modulo 7 to accommodate at least 6 packets.

Linear equations:

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

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

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

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

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

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

Linear equations:

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

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

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

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

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

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

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

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

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

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

Send
Example

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

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

\[
6a_1 + 3a_0 = 2 \quad (\text{mod } 7), \quad 5a_1 + 4a_0 = 0 \quad (\text{mod } 7)
\]

\[
a_1 = 2a_0, \quad a_0 = 2 \quad (\text{mod } 7) \quad a_1 = 4 \quad (\text{mod } 7) \quad a_2 = 2 \quad (\text{mod } 7)
\]

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

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

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

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

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

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

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

Send

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

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) (2,4), (6,0)
Bad reception!

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

Receive: (1,1), (2,4), (6,0)
Reconstruct?
Bad reception!

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

Receive: (1,1) (2,4), (6,0)

Reconstruct?

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

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

Recieve: \((1,1)\) (2,4), (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) (2,4), (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)

Recieve: (1,1) (2,4), (6,0)
   Reconstruct?

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

Lagrange or linear equations.

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

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

Recieve: \((1,1) \ (2,4), \ (6,0)\)
  
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) (2, 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: \((1,1)\) \((2,4)\), \((6,0)\)
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) (2, 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) (2,4), (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) (2,4), (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 ...

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

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

Recieve: (1, 1) (2, 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) (2,4), (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) (2,4), (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) (2, 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? \(P(1) = 1, P(2) = 4, P(3) = 4\).
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
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!

The other constraint: arithmetic system can represent 0, 1, 2, 3, 4.

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!

Remember the secret, $s = 144$, must be one of the possible values.
You want to encode a secret consisting of 1,4,4.

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

Remember the secret, $s = 144$, must be one of the possible values.

You want to send a message consisting of packets 1,4,2,3,0
You want to encode a secret consisting of 1, 4, 4.

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

Remember the secret, $s = 144$, must be one of the possible values.

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

You want to encode a secret consisting of 1,4,4.
   How big should modulus be?
      Larger than 144 and prime!
   Remember the secret, $s = 144$, must be one of the possible values.
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?
Questions for Review

You want to encode a secret consisting of 1,4,4.
   How big should modulus be?
      Larger than 144 and prime!
   Remember the secret, $s = 144$, must be one of the possible values.
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!

Remember the secret, $s = 144$, must be one of the possible values.

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!

Remember the secret, \( s = 144 \), must be one of the possible values.

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!
   Remember the secret, $s = 144$, must be one of the possible values.

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!
   The other constraint: arithmetic system can represent 0,1,2,3,4.
Questions for Review

You want to encode a secret consisting of 1,4,4.
   How big should modulus be?
      Larger than 144 and prime!
   Remember the secret, \( s = 144 \), must be one of the possible values.

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!
   The other constraint: arithmetic system can represent 0, 1, 2, 3, 4.

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!

Remember the secret, \( s = 144 \), must be one of the possible values.

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!

The other constraint: arithmetic system can represent 0, 1, 2, 3, 4.

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: corruptions $k$ 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

GPS device

3 packet message.
Error Correction

Satellite

GPS device

3 packet message.

Corrupts 1 packets.
Error Correction

3 packet message. Send 5.

Corrupts 1 packets.

Satellite

GPS device
Error Correction

3 packet message. Send 5.

Corrupts 1 packets.
Error Correction

Satellite

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:**
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.

   $\quad P(1) = m_1, \ldots, P(n) = m_n.$
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:** Recive 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.

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

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

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

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

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

\( P(x) \): 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.
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.
(2) Degree \( n - 1 \) polynomial \( Q(x) \) consistent with \( n + k \) points.
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 \( k \)’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.

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

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

\[ P(x): \text{ degree } n - 1 \text{ polynomial.} \]
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.
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 : \( n+2k \).
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.
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.
   $\implies Q(i) = P(i)$ 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: \( 2n + 2k \).
Total points to choose from: \( n + 2k \).
Points contained by both: \( \geq n \).
\[ \geq P - H \] Collisions.
\[ \implies Q(i) = P(i) \text{ at } n \text{ points.} \]
\[ \implies Q(x) = P(x). \]
Example.

Message: 3, 0, 6.
Example.

Message: 3, 0, 6.

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

Message: 3, 0, 6.

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

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 $P(1) = 3, P(2) = 0, P(3) = 6 \pmod{7}$.

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: \( P(x) = x^2 + x + 1 \) (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 \( R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3 \).
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 $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$.

$P(i) = R(i)$ 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.
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.
**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
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 \( P(x) = p_2x^2 + p_1x + p_0 \) 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..

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

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

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..

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

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..

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

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.

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

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

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

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..

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

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..

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

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

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

\[ p_{n-1} + \cdots p_0 \equiv R(1) \pmod{p} \]
\[ p_{n-1}2^{n-1} + \cdots p_0 \equiv R(2) \pmod{p} \]
\[ \vdots \]
\[ p_{n-1}i^{n-1} + \cdots p_0 \equiv R(i) \pmod{p} \]
\[ \vdots \]
\[ p_{n-1}(m)^{n-1} + \cdots p_0 \equiv R(m) \pmod{p} \]
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!!
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???
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!!!
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.
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 \( R(1), \ldots, R(m = n + 2k) \).

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

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

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

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

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

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

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

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...

Oh where, Oh where

Oh where, Oh where
Ditty...

Oh where, Oh where
has my little dog gone?

Oh where, oh where can he be
With his ears cut short
And his tail cut long
Oh where, oh where can he be?

Oh where, Oh where
have my packets gone...
Oh where, oh where do they not fit.
With the polynomial well put
But the channel a bit wrong
Where, oh where do we look?
Ditty...

Oh where, Oh where
has my little dog gone?
Oh where, oh where can he be
Ditty...

Oh where, Oh where
has my little dog gone?
Oh where, oh where can he be
With his ears cut short

wrong?

Oh where, oh where do they not fit.
With the polynomial well put
But the channel a bit wrong
Where, oh where do we look?
Ditty...

Oh where, Oh where has my little dog gone?
Oh where, oh where can he be

With his ears cut short
And his tail cut long
Ditty...

Oh where, Oh where
has my little dog gone?
Oh where, oh where can he be

With his ears cut short
And his tail cut long
Oh where, oh where can he be?
Ditty...

Oh where, Oh where
has my little dog gone?
Oh where, oh where can he be

With his ears cut short
And his tail cut long
Oh where, oh where can he be?

Oh where, Oh where
Oh where, Oh where
has my little dog gone?
Oh where, oh where can he be

With his ears cut short
And his tail cut long
Oh where, oh where can he be?

Oh where, Oh where
Oh where, Oh where
has my little dog gone?
Oh where, oh where can he be

With his ears cut short
And his tail cut long
Oh where, oh where can he be?

Oh where, Oh where
have my packets gone..
Ditty...

Oh where, Oh where has my little dog gone?
Oh where, oh where can he be

With his ears cut short
And his tail cut long
Oh where, oh where can he be?

Oh where, Oh where have my packets gone.. wrong?
Oh where, Oh where
has my little dog gone?
Oh where, oh where can he be
With his ears cut short
And his tail cut long
Oh where, oh where can he be?

Oh where, Oh where
have my packets gone.. wrong?
Oh where, oh where do they not fit.
Oh where, Oh where
has my little dog gone?
Oh where, oh where can he be

With his ears cut short
And his tail cut long
Oh where, oh where can he be?

Oh where, Oh where
have my packets gone.. wrong?
Oh where, oh where do they not fit.
With the polynomial well put
Ditty...

Oh where, Oh where has my little dog gone?
Oh where, oh where can he be
With his ears cut short
And his tail cut long
Oh where, oh where can he be?

Oh where, Oh where have my packets gone... wrong?
Oh where, oh where do they not fit.
With the polynomial well put
But the channel a bit wrong
Oh where, Oh where
has my little dog gone?
Oh where, oh where can he be

With his ears cut short
And his tail cut long
Oh where, oh where can he be?

Oh where, Oh where
have my packets gone.. wrong?
Oh where, oh where do they not fit.

With the polynomial well put
But the channel a bit wrong
Where, oh where do we look?
Where oh where can my bad packets be?

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

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

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

\[\vdots\]

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

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

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

\[
\vdots
\]

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

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

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

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

\[\vdots\]

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

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

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

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\). Zero times anything is zero!!!! My love is won.
Where oh where can my bad packets be?

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

Idea: Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\). Zero times anything is zero!!!!! My love is won. All equations satisfied!!!!!
Where oh where can my bad packets be?

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

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\).
Zero times anything is zero!!!! My love is won.
All equations satisfied!!!!!

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

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

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

\[
\vdots
\]

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

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

Zero times anything is zero!!!!! My love is won.

All equations satisfied!!!!!

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

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

Idea: Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\).
Zero times anything is zero!!!! My love is won.
All equations satisfied!!!!!

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

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

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\).
Zero times anything is zero!!!! My love is won.
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?

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

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

Zero times anything is zero!!!!! My love is won.

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?

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

**Idea:** Multiply equation \( i \) by 0 if and only if \( P(i) \neq R(i) \).
- Zero times anything is zero!!!! My love is won.
- 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**?

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

**Idea:** Multiply equation \( i \) by 0 if and only if \( P(i) \neq R(i) \).
Zero times anything is zero!!!!! My love is won.
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?

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

$$(p_{n-1}2^{n-1} + \cdots + p_0) \equiv R(2) \pmod{p}$$

$$\vdots$$

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

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

Zero times anything is zero!!!!! My love is won.

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?

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

**Idea:** Multiply equation \( i \) by 0 if and only if \( P(i) \neq R(i) \). Zero times anything is zero!!!! My love is won. 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?

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

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\). Zero times anything is zero!!!! My love is won. 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:** \(E(x) = (x - e_1)(x - e_2)\ldots\)
Where oh where can my bad packets be?

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

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\).
Zero times anything is zero!!!! My love is won.
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?

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

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

Zero times anything is zero!!!! My love is won.

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 \(P(i) \neq R(i)\).

Zero times anything is zero!!!! My love is won.

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:** \(E(x) = (x - e_1)(x - e_2)\ldots(x - e_k)\).

\(E(i) = 0\) 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 \( P(i) \neq R(i) \).

Zero times anything is zero!!!!! My love is won.

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:** \( E(x) = (x - e_1)(x - e_2)\ldots(x - e_k) \).

\( E(i) = 0 \) 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?

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

**Idea:** Multiply equation \( i \) by 0 if and only if \( P(i) \neq R(i) \).
Zero times anything is zero!!!!! My love is won.
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...

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

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

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

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

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

\[
(p_2 + p_1 + p_0) \equiv (3) \pmod{7} \\
(4p_2 + 2p_1 + p_0) \equiv (1) \pmod{7} \\
(2p_2 + 3p_1 + p_0) \equiv (6) \pmod{7} \\
(2p_2 + 4p_1 + p_0) \equiv (0) \pmod{7} \\
(4p_2 + 5p_1 + p_0) \equiv (3) \pmod{7}
\]

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...

\[
(1 - 2)(p_2 + p_1 + p_0) \equiv (3)(1 - 2) \pmod{7}
\]
\[
(2 - 2)(4p_2 + 2p_1 + p_0) \equiv (1)(2 - 2) \pmod{7}
\]
\[
(3 - 2)(2p_2 + 3p_1 + p_0) \equiv (6)(3 - 2) \pmod{7}
\]
\[
(4 - 2)(2p_2 + 4p_1 + p_0) \equiv (0)(4 - 2) \pmod{7}
\]
\[
(5 - 2)(4p_2 + 5p_1 + p_0) \equiv (3)(5 - 2) \pmod{7}
\]

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...

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

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

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

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

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

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...

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

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...

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

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

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

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

\[
(1 - 2)(p_2 + p_1 + p_0) \equiv (3)(1 - 2) \pmod 7 \\
(2 - 2)(4p_2 + 2p_1 + p_0) \equiv (1)(2 - 2) \pmod 7 \\
(3 - 2)(2p_2 + 3p_1 + p_0) \equiv (6)(3 - 2) \pmod 7 \\
(4 - 2)(2p_2 + 4p_1 + p_0) \equiv (0)(4 - 2) \pmod 7 \\
(5 - 2)(4p_2 + 5p_1 + p_0) \equiv (3)(5 - 2) \pmod 7
\]

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...

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

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

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

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

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.

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: $(x - e)$.

4 unknowns ($p_0, p_1, p_2$ and $e$), 5 nonlinear equations.
..turn their heads each day,

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

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

\[ \vdots \]

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

\[ \vdots \]

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

...so satisfied, I’m on my way.

\[ m = n + 2k \] satisfied equations,
..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,

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

...so satisfied, I’m on my way.

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

\[ m = n + 2k \] 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.

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

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,

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

...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 $k$ ...

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

\[ \implies k \text{ (unknown) coefficients.} \]
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.
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 $n + k - 1$
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 $n + k - 1$ ...

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

- $E(x)$ has degree $k$ ...
  \[
  E(x) = x^k + b_{k-1}x^{k-1} \cdots b_0.
  \]
  $\Rightarrow$ $k$ (unknown) coefficients. Leading coefficient is 1.

- $Q(x) = P(x)E(x)$ has degree $n + k - 1$ ...
  \[
  Q(x) = a_{n+k-1}x^{n+k-1} + a_{n+k-2}x^{n+k-2} + \cdots a_0
  \]
  $\Rightarrow$ $n + k$ (unknown) coefficients.

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

\begin{itemize}
  \item \( E(x) \) has degree \( k \) ...

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

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

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

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

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

Number of unknown coefficients: \( n + 2k \).
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.

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

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

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

\vdots
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.

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

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

$$\vdots$$

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

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

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

$$\vdots$$

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

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

Solve for coefficients of $Q(x)$ and $E(x)$. 
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 $1, \ldots, i, n+2k = m,$

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

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.

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

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

$Q(x) = E(x)P(x) = a_3 x^3 + a_2 x^2 + a_1 x + a_0$
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 $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$

$Q(x) = E(x)P(x) = a_3x^3 + a_2x^2 + a_1x + a_0$

$E(x) = x - b_0$

$Q(i) = R(i)E(i)$.

$$
\begin{align*}
    a_3 + a_2 + a_1 + a_0 & \equiv 3(1 - b_0) \pmod{7} \\
    a_3 + 4a_2 + 2a_1 + a_0 & \equiv 1(2 - b_0) \pmod{7} \\
    6a_3 + 2a_2 + 3a_1 + a_0 & \equiv 6(3 - b_0) \pmod{7} \\
    a_3 + 2a_2 + 4a_1 + a_0 & \equiv 0(4 - b_0) \pmod{7} \\
    6a_3 + 4a_2 + 5a_1 + a_0 & \equiv 3(5 - b_0) \pmod{7}
\end{align*}
$$
Example.

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

\( Q(x) = E(x)P(x) = a_3x^3 + a_2x^2 + a_1x + a_0 \)

\( E(x) = x - b_0 \)

\( Q(i) = R(i)E(i). \)

\[
\begin{align*}
    a_3 + a_2 + a_1 + a_0 & \equiv 3(1 - b_0) \pmod{7} \quad \\
    a_3 + 4a_2 + 2a_1 + a_0 & \equiv 1(2 - b_0) \pmod{7} \quad \\
    6a_3 + 2a_2 + 3a_1 + a_0 & \equiv 6(3 - b_0) \pmod{7} \quad \\
    a_3 + 2a_2 + 4a_1 + a_0 & \equiv 0(4 - b_0) \pmod{7} \quad \\
    6a_3 + 4a_2 + 5a_1 + a_0 & \equiv 3(5 - b_0) \pmod{7} 
\end{align*}
\]

\( a_3 = 1, \ a_2 = 6, \ a_1 = 6, \ a_0 = 5 \) and \( b_0 = 2. \)
Example.

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

$Q(x) = E(x)P(x) = a_3 x^3 + a_2 x^2 + a_1 x + a_0$

$E(x) = x - b_0$

$Q(i) = R(i)E(i)$.

\[
\begin{align*}
  a_3 + a_2 + a_1 + a_0 & \equiv 3(1 - b_0) \pmod 7 \\
  a_3 + 4a_2 + 2a_1 + a_0 & \equiv 1(2 - b_0) \pmod 7 \\
  6a_3 + 2a_2 + 3a_1 + a_0 & \equiv 6(3 - b_0) \pmod 7 \\
  a_3 + 2a_2 + 4a_1 + a_0 & \equiv 0(4 - b_0) \pmod 7 \\
  6a_3 + 4a_2 + 5a_1 + a_0 & \equiv 3(5 - b_0) \pmod 7
\end{align*}
\]

$a_3 = 1, a_2 = 6, a_1 = 6, a_0 = 5$ 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.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{r@{}c@{}l}
x - 2 & ) & x^3 + 6x^2 + 6x + 5 \\
& & \hline \\
& & x^3 - 2x^2 \\
& & \hline \\
& & 8x^2 + 6x \\
& & 8x^2 - 16x \\
& & \hline \\
& & 22x + 5 \\
& & 22x - 44 \\
& & \hline \\
& & 49 
\end{array}
\]

\[ P(x) = x^2 + x + 1 \]

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.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{r}
1 & x^2 \\
\hline
x - 2 & x^3 + 6x^2 + 6x + 5 \\
\hline
 & x^3 - 2x^2 \\
\end{array}
\]

\[ P(x) = x^2 + x + 1 \]

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.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{c}
\frac{1}{x-2} \\
\hline
x^3 + 6x^2 + 6x + 5 \\
\hline
x^3 - 2x^2 \\
\hline
1x^2 + 6x + 5
\end{array}
\]

Message is \( P(1) = 3, P(2) = 0, P(3) = 6. \)

What is \( x^2 - 2x - 2 \) except at \( x = 2 \)?

Hole there?
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{llllllll}
1 & x^2 & + & 1 & x & \hline \\
\hline
x - 2 & ) & x^3 & + & 6x^2 & + & 6x & + & 5 \\
\hline
x^3 & - & 2x^2 & & & & & \\
\hline
1 & x^2 & + & 6x & + & 5 \\
1 & x^2 & - & 2x & & & & \\
\hline
\end{array}
\]
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{cccc}
1 & x^2 & + & 1 & x \\
\hline
x - 2 & ) & x^3 & + & 6x^2 & + & 6x & + & 5 \\
\hline & x^3 & - & 2x^2 & & & & & \\
\hline & 1 & x^2 & + & 6x & + & 5 \\
1 & x^2 & - & 2x & & & & & \\
\hline & x & + & 5
\end{array}
\]
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{r}
\text{1} & x^2 & + & 1 & x & + & 1 \\
\hline
\text{x - 2} & | & x^3 & + & 6x^2 & + & 6x & + & 5 \\
\text{x^3} & - & 2x^2 & \hline
\text{1} & x^2 & + & 6x & + & 5 \\
\text{1} & x^2 & - & 2x & \hline
\text{x + 5} \\
\text{x - 2}
\end{array}
\]

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.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{r}
\begin{align*}
1 & \quad x^2 & + & 1 & \quad x & + & 1 \\
\hline
x - 2 & \quad ) & x^3 & + & 6x^2 & + & 6x & + & 5 \\
& & x^3 & - & 2x^2 & & & & \\
\hline
& & 1x^2 & + & 6x & + & 5 \\
& & 1x^2 & - & 2x & & & & \\
\hline
& & x & + & 5 & \end{align*}
\end{array}
\]

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.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{c}
1 \ x^2 + 1 \ x + 1 \\
\hline
x - 2 \ ) \ x^3 + 6 \ x^2 + 6 \ x + 5 \\
\hline
x^3 - 2 \ x^2 \\
\hline
1 \ x^2 + 6 \ x + 5 \\
1 \ x^2 - 2 \ x \\
\hline
x + 5 \\
x - 2 \\
\hline
0
\end{array}
\]

\[ P(x) = x^2 + x + 1 \]
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{c}
1 \ x^2 + 1 \ x + 1 \\
\hline
x - 2 \\
\end{array}
\]

\[
\begin{array}{c}
x^3 + 6x^2 + 6x + 5 \\
x^3 - 2x^2 \\
\hline
1 \ x^2 + 6x + 5 \\
1 \ x^2 - 2x \\
\hline
x + 5 \\
x - 2 \\
\hline
0
\end{array}
\]

\[ P(x) = x^2 + x + 1 \]

Message is \( P(1) = 3, P(2) = 0, P(3) = 6. \)
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{cccc}
1 & x^2 & + & 1 & x & + & 1 \\
\hline \\
x - 2 & ) & x^3 & + & 6x^2 & + & 6x & + & 5 \\
& x^3 & - & 2x^2 & & & & & \\
\hline \\
& 1 & x^2 & + & 6x & + & 5 \\
\end{array}
\]

\[
\begin{array}{cccc}
1 & x^2 & - & 2x & & & & & \\
\hline \\
& x & + & 5 \\
\end{array}
\]

\[ P(x) = x^2 + x + 1 \]
Message is \( P(1) = 3, P(2) = 0, P(3) = 6. \)
What is \( \frac{x-2}{x-2} \)?
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{c}
1 \ x^2 + 1 \ x + 1 \\
\hline \\
\end{array}
\]

\[
\begin{array}{c}
x - 2 \ ) x^3 + 6 \ x^2 + 6 \ x + 5 \\
\hline \\
x^3 - 2 \ x^2 \\
\hline \\
1 \ x^2 + 6 \ x + 5 \\
\hline \\
1 \ x^2 - 2 \ x \\
\hline \\
x + 5 \\
x - 2 \\
\hline \\
0 \\
\end{array}
\]

\[ P(x) = x^2 + x + 1 \]

Message is \( P(1) = 3, P(2) = 0, P(3) = 6. \)

What is \( \frac{x-2}{x-2} \)? 1
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[ \begin{array}{r}
1 & x^2 & + & 1 & x & + & 1 \\
\hline
x - 2 & ) & x^3 & + & 6 & x^2 & + & 6 & x & + & 5 \\
\hline
x^3 & - & 2 & x^2 & & \\
\hline
1 & x^2 & + & 6 & x & + & 5 \\
1 & x^2 & - & 2 & x & \\
\hline
x & + & 5 \\
x & - & 2 \\
\hline
0
\end{array} \]

\[ P(x) = x^2 + x + 1 \]
Message is \( P(1) = 3, P(2) = 0, P(3) = 6. \)

What is \( \frac{x-2}{x-2} \)? 1

Except at \( x = 2 \)?
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{c|cccc}
& x^3 & + 6x^2 & + 6x & + 5 \\
\hline
x - 2 & x^3 & - 2x^2 \\
\hline
& x^2 & + 6x & + 5 \\
& x^2 & - 2x \\
\hline
& x & + 5 \\
& x & - 2 \\
\hline
& 0
\end{array}
\]

\[ P(x) = x^2 + x + 1 \]
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.
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.
Is there one and only one $P(x)$ from Berlekamp-Welsh procedure?
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

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).$$  \hspace{1cm} (1)

**Proof:**

...
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).$$

(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

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).$$  \hfill (1)

**Proof:**

We claim

$$Q'(x)E(x) = Q(x)E'(x) \text{ on } n + 2k \text{ values of } x.$$  \hfill (2)

Equation 2 implies 1:
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

\[
\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).
\]  
(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 $n + 2k - 1$
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.
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.
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x). \tag{1}$$

**Proof:**
We claim

$$Q'(x)E(x) = Q(x)E'(x) \text{ on } n+2k \text{ values of } x. \tag{2}$$

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

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x). \quad (1)$$

**Proof:**

We claim

$$Q'(x)E(x) = Q(x)E'(x) \text{ on } n + 2k \text{ values of } x. \quad (2)$$

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.

$$\implies \frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} \text{ equal on } n \text{ points.}$$
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.

$$\Rightarrow \frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} \text{ equal on } n \text{ points.}$$

Both degree $\leq n-1$
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

$$Q'(x)E(x) = Q(x)E'(x) \text{ on } n+2k \text{ values of } x. \quad (2)$$

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-1 \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 $\leq n - 1 \implies$ Same polynomial!
Last bit.

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:
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 $Q(i) = 0$. 

Cross multiplying gives equality in fact for these points. Points to polynomials, have to deal with zeros!

Example: dealing with $x^2 - 2x - 2$ at $x = 2$. 

Last bit.
**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 $Q'(i) = 0$. 

Cross multiplying gives equality in fact for these points.

Points to polynomials, have to deal with zeros!

Example: dealing with $x - 2$ at $x = 2$. 
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 \( Q'(i) = 0 \).

\[ \implies Q(i)E'(i) = Q'(i)E(i) \text{ holds when } E(i) \text{ or } E'(i) \text{ are zero.} \]

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.
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 \( Q'(i) = 0 \).

\[\Rightarrow\] \( Q(i)E'(i) = Q'(i)E(i) \) 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 \( Q'(i) = 0 \).

\[ \implies Q(i)E'(i) = Q'(i)E(i) \] 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!
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!
Poll

Say you sent a message of length 4, encoded as \( P(x) \) where one sends packets \( P(1), \ldots, P(8) \).

You receive packets \( R(1), \ldots, R(8) \).

Packets 1 and 4 are corrupted.

(A) \( R(1) \neq P(1) \)
(B) The degree of \( P(x)E(x) = 3 + 2 = 5 \).
(C) The degree of \( E(x) \) is 2.
(D) The number of coefficients of \( P(x) \) is 4.
(E) The number of coefficients of \( P(x)Q(x) \) is 6.
Say you sent a message of length 4, encoded as $P(x)$ where one sends packets $P(1), \ldots P(8)$.

You receive packets $R(1), \ldots R(8)$.

Packets 1 and 4 are corrupted.

(A) $R(1) \neq P(1)$
(B) The degree of $P(x)E(x) = 3 + 2 = 5$.
(C) The degree of $E(x)$ is 2.
(D) The number of coefficients of $P(x)$ is 4.
(E) The number of coefficients of $P(x)Q(x)$ is 6.

(E) is false.
Say you sent a message of length 4, encoded as $P(x)$ where one sends packets $P(1),...P(8)$.

You receive packets $R(1),...R(8)$.

Packets 1 and 4 are corrupted.

(A) $R(1) \neq P(1)$
(B) The degree of $P(x)E(x) = 3 + 2 = 5$.
(C) The degree of $E(x)$ is 2.
(D) The number of coefficients of $P(x)$ is 4.
(E) The number of coefficients of $P(x)Q(x)$ is 6.

(E) is false.

(A) $E(x) = (x - 1)(x - 4)$
(B) The number of coefficients in $E(x)$ is 2.
(C) The number of unknown coefficients in $E(x)$ is 2.
(D) $E(x) = (x - 1)(x - 2)$
(E) $R(4) \neq P(4)$
(F) The degree of $R(x)$ is 5.
Poll

Say you sent a message of length 4, encoded as $P(x)$ where one sends packets $P(1),\ldots,P(8)$.

You receive packets $R(1),\ldots,R(8)$.

Packets 1 and 4 are corrupted.

(A) $R(1) \neq P(1)$
(B) The degree of $P(x)E(x) = 3 + 2 = 5$.
(C) The degree of $E(x)$ is 2.
(D) The number of coefficients of $P(x)$ is 4.
(E) The number of coefficients of $P(x)Q(x)$ is 6.

(E) is false.

(A) $E(x) = (x - 1)(x - 4)$
(B) The number of coefficients in $E(x)$ is 2.
(C) The number of unknown coefficients in $E(x)$ is 2.
(D) $E(x) = (x - 1)(x - 2)$
(E) $R(4) \neq P(4)$
(F) The degree of $R(x)$ is 5.

(A), (C), (E). (F) doesn’t type check!
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets?
Summary. Error Correction.

Communicate \( n \) packets, with \( k \) erasures.

How many packets? \( n + k \)
Summary. Error Correction.

Communicate \( n \) packets, with \( k \) erasures.

How many packets? \( n + k \)
How to encode?
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$. 
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree?
Summary. Error Correction.

Communicate \( n \) packets, with \( k \) erasures.

How many packets? \( n + k \)
How to encode? With polynomial, \( P(x) \).
Of degree? \( n - 1 \)
Summary. 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?
Summary. 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!
Summary. 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.
Summary. 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?
Summary. 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? $n + 2k$
Summary. 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?
Summary. 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
Summary. 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?
Summary. 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, \( P(x) \).
Summary. 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?

Reed-Solomon codes. Welsh-Berlekamp Decoding. Perfection!
Summary. 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? \( n - 1 \).
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?
Summary. 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?
Summary. 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 $P(x)$!
Summary. 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.
Summary. 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 $Q(x) = E(x)P(x)$. 
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.
Summary. 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!
Summary. 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! \( P(x) = Q(x)/E(x) \)!
Summary. 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.
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.
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. Perfection!
Cool.

Really Cool!