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!

In general..

Given points: $(x_1, y_1); (x_2, y_2) \cdots (x_k, y_k)$.

In general..

Given points: $(x_1, y_1); (x_2, y_2) \cdots (x_k, y_k)$.

Solve...

$$\begin{aligned}
a_{k-1}x_1^{k-1} + \cdots + a_0 &\equiv y_1 \pmod{p} \\
a_{k-1}x_2^{k-1} + \cdots + a_0 &\equiv y_2 \pmod{p} \\
\vdots \quad &\vdots \quad \vdots \\
a_{k-1}x_k^{k-1} + \cdots + a_0 &\equiv y_k \pmod{p}
\end{aligned}$$

In general..

Given points: $(x_1, y_1); (x_2, y_2) \cdots (x_k, y_k)$.

Solve...

$$\begin{aligned}
a_{k-1}x_1^{k-1} + \cdots + a_0 &\equiv y_1 \pmod p \\
a_{k-1}x_2^{k-1} + \cdots + a_0 &\equiv y_2 \pmod p \\
\vdots \quad & \quad \vdots \quad \vdots \\
a_{k-1}x_k^{k-1} + \cdots + a_0 &\equiv y_k \pmod p
\end{aligned}$$

Will this always work?

In general..

Given points: $(x_1, y_1); (x_2, y_2) \cdots (x_k, y_k)$.

Solve...

$$\begin{aligned}
a_{k-1}x_1^{k-1} + \cdots + a_0 &\equiv y_1 \pmod{p} \\
a_{k-1}x_2^{k-1} + \cdots + a_0 &\equiv y_2 \pmod{p} \\
\vdots \quad &\vdots \quad \vdots \\
a_{k-1}x_k^{k-1} + \cdots + a_0 &\equiv y_k \pmod{p}
\end{aligned}$$

Will this always work?

As long as solution **exists** and it is **unique!** And...

In general..

Given points: $(x_1, y_1); (x_2, y_2) \cdots (x_k, y_k)$.

Solve...

$$
\begin{aligned}
a_{k-1}x_1^{k-1} + \cdots + a_0 &\equiv y_1 \pmod{p} \\
a_{k-1}x_2^{k-1} + \cdots + a_0 &\equiv y_2 \pmod{p} \\
\vdots \qquad & \vdots \quad \vdots \\
a_{k-1}x_k^{k-1} + \cdots + a_0 &\equiv y_k \pmod{p}
\end{aligned}
$$

Will this always work?

As long as solution **exists** and it is **unique!** And...

In general..

Given points: $(x_1, y_1); (x_2, y_2) \cdots (x_k, y_k)$.

Solve...

$$
\begin{aligned}
a_{k-1} x_1^{k-1} + \cdots + a_0 &\equiv y_1 \pmod{p} \\
a_{k-1} x_2^{k-1} + \cdots + a_0 &\equiv y_2 \pmod{p} \\
\vdots \qquad & \vdots \qquad \vdots \\
a_{k-1} x_k^{k-1} + \cdots + a_0 &\equiv y_k \pmod{p}
\end{aligned}
$$

Will this always work?

As long as solution **exists** and it is **unique!** And...

Modular Arithmetic Fact: Exactly 1 polynomial of degree $\leq d$ with arithmetic modulo prime p contains $d+1$ pts.

Uniqueness.

Modular Arithmetic Fact: Exactly 1 polynomial of degree $\leq d$ with arithmetic modulo prime p contains $d+1$ pts.

Uniqueness.

Modular Arithmetic Fact: Exactly 1 polynomial of degree $\leq d$ with arithmetic modulo prime p contains $d+1$ pts.

Existence: Lagrange interpolation.

Uniqueness.

Modular Arithmetic Fact: Exactly 1 polynomial of degree $\leq d$ with arithmetic modulo prime p contains $d + 1$ pts.

Existence: Lagrange interpolation. Uniqueness?

Uniqueness.

Modular Arithmetic Fact: Exactly 1 polynomial of degree $\leq d$ with arithmetic modulo prime p contains $d+1$ pts.

Existence: Lagrange interpolation. Uniqueness?

Uniqueness Fact. At most one degree d polynomial hits $d+1$ points.

Uniqueness Fact. At most one degree d polynomial contains $d+1$ points.

Uniqueness Fact. At most one degree d polynomial contains $d+1$ points.

Proof:

Uniqueness Fact. At most one degree d polynomial contains $d+1$ points.

Proof:

Roots fact: Any degree d polynomial has at most d roots.

Uniqueness Fact. At most one degree d polynomial contains $d+1$ points.

Proof:

Roots fact: Any degree d polynomial has at most d roots.

Assume two different polynomials $Q(x)$ and $P(x)$ hits $d+1$ points.

Uniqueness Fact. At most one degree d polynomial contains $d+1$ points.

Proof:

Roots fact: Any degree d polynomial has at most d roots.

Assume two different polynomials $Q(x)$ and $P(x)$ hits $d+1$ points.

$R(x) = Q(x) - P(x)$ has $d+1$ roots and is degree d.

Uniqueness Fact. At most one degree d polynomial contains $d+1$ points.

Proof:

Roots fact: Any degree d polynomial has at most d roots.

Assume two different polynomials $Q(x)$ and $P(x)$ hits $d+1$ points.

$R(x) = Q(x) - P(x)$ has $d+1$ roots and is degree d.
Contradiction.

Uniqueness Fact. At most one degree d polynomial contains $d+1$ points.

Proof:

Roots fact: Any degree d polynomial has at most d roots.

Assume two different polynomials $Q(x)$ and $P(x)$ hits $d+1$ points.

$R(x) = Q(x) - P(x)$ has $d+1$ roots and is degree d.
Contradiction. $\square$

Uniqueness Fact. At most one degree d polynomial contains $d+1$ points.

Proof:

Roots fact: Any degree d polynomial has at most d roots.

Assume two different polynomials $Q(x)$ and $P(x)$ hits $d+1$ points.

$R(x) = Q(x) - P(x)$ has $d+1$ roots and is degree d.
Contradiction. $\square$
Must prove **Roots fact.**

Polynomial Division.

Divide $4x^2 - 3x + 2$ by $(x - 3)$ modulo 5.

Polynomial Division.

Divide $4x^2 - 3x + 2$ by $(x - 3)$ modulo 5.

```
                    4 x
            -----------------
x - 3 ) 4x^2 - 3 x + 2
```

Polynomial Division.

Divide $4x^2 - 3x + 2$ by $(x - 3)$ modulo 5.

```
                  4 x
         -----------------
x - 3 )  4x^2 - 3 x + 2
         4x^2 - 2x
```

Polynomial Division.

Divide $4x^2 - 3x + 2$ by $(x - 3)$ modulo 5.

```
              4 x + 4
         -----------------
x - 3 )  4x^2 - 3 x + 2
         4x^2 - 2x
         ----------
               4x + 2
```

Polynomial Division.

Divide $4x^2 - 3x + 2$ by $(x - 3)$ modulo 5.

```
                4 x + 4
         -----------------
x - 3 )  4x^2 - 3 x + 2
         4x^2 - 2x
         ----------
                 4x + 2
                 4x - 2
```

Polynomial Division.

Divide $4x^2 - 3x + 2$ by $(x - 3)$ modulo 5.

```
                  4 x + 4
         -----------------
x - 3 )  4x^2 - 3 x + 2
         4x^2 - 2x
         ----------
                 4x + 2
                 4x - 2
                 -------
                      4
```

Polynomial Division.

Divide $4x^2 - 3x + 2$ by $(x - 3)$ modulo 5.

```
                4 x + 4 r 4
         -----------------
x - 3 )  4x^2 - 3 x + 2
         4x^2 - 2x
         ----------
                 4x + 2
                 4x - 2
                 -------
                      4
```

Polynomial Division.

Divide $4x^2 - 3x + 2$ by $(x - 3)$ modulo 5.

```
                 4 x + 4 r 4
         -----------------
x - 3 )  4x^2 - 3 x + 2
         4x^2 - 2x
         ----------
                 4x + 2
                 4x - 2
                 -------
                      4
```

$4x^2 - 3x + 2 \equiv (x - 3)(4x + 4) + 4 \pmod 5$

Polynomial Division.

Divide $4x^2 - 3x + 2$ by $(x - 3)$ modulo 5.

```
             4 x + 4 r 4
        -----------------
x - 3 ) 4x^2 - 3 x + 2
        4x^2 - 2x
        ----------
                4x + 2
                4x - 2
                -------
                     4
```

$4x^2 - 3x + 2 \equiv (x - 3)(4x + 4) + 4 \pmod 5$

In general, divide $P(x)$ by $(x - a)$ gives $Q(x)$ and remainder r.

Polynomial Division.

Divide $4x^2 - 3x + 2$ by $(x - 3)$ modulo 5.

```
                4 x + 4 r 4
         -----------------
x - 3 )  4x^2 - 3 x + 2
         4x^2 - 2x
         ----------
                 4x + 2
                 4x - 2
                 -------
                      4
```

$4x^2 - 3x + 2 \equiv (x - 3)(4x + 4) + 4 \pmod 5$

In general, divide $P(x)$ by $(x - a)$ gives $Q(x)$ and remainder r.

That is, $P(x) = (x - a)Q(x) + r$

Polynomial Division.

Divide $4x^2 - 3x + 2$ by $(x - 3)$ modulo 5.

```
              4 x + 4 r 4
         -----------------
x - 3 )  4x^2 - 3 x + 2
         4x^2 - 2x
         ----------
                4x + 2
                4x - 2
                -------
                     4
```

$4x^2 - 3x + 2 \equiv (x-3)(4x+4) + 4 \pmod 5$

In general, divide $P(x)$ by $(x - a)$ gives $Q(x)$ and remainder r.

That is, $P(x) = (x - a)Q(x) + r$

r is degree 0 polynomial..

Polynomial Division.

Divide $4x^2 - 3x + 2$ by $(x-3)$ modulo 5.

```
                4 x + 4 r 4
         -----------------
x - 3 )  4x^2 - 3 x + 2
         4x^2 - 2x
         ----------
                4x + 2
                4x - 2
                -------
                     4
```

$4x^2 - 3x + 2 \equiv (x-3)(4x+4) + 4 \pmod 5$

In general, divide $P(x)$ by $(x-a)$ gives $Q(x)$ and remainder r.

That is, $P(x) = (x-a)Q(x) + r$

r is degree 0 polynomial..or a constant!

Only d roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0: $P(x) = (x-a)Q(x)$.

Only d roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0: $P(x) = (x-a)Q(x)$.

Proof: $P(x) = (x-a)Q(x) + r$.

Only d roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0:
$P(x) = (x-a)Q(x)$.

Proof: $P(x) = (x-a)Q(x) + r$.
Plugin a:

Only d roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0:
$P(x) = (x-a)Q(x)$.

Proof: $P(x) = (x-a)Q(x) + r$.
Plugin a: $P(a)$

Only d roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0: $P(x) = (x-a)Q(x)$.

Proof: $P(x) = (x-a)Q(x) + r$.
Plugin a: $P(a) = r$

Only d roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0:
$P(x) = (x-a)Q(x)$.

Proof: $P(x) = (x-a)Q(x) + r$.
Plugin a: $P(a) = r = 0$.

Only d roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0: $P(x) = (x-a)Q(x)$.

Proof: $P(x) = (x-a)Q(x) + r$.
Plugin a: $P(a) = r = 0$. □

Only d roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0:
$P(x) = (x-a)Q(x)$.

Proof: $P(x) = (x-a)Q(x) + r$.
Plugin a: $P(a) = r = 0$. □

Lemma 2: $P(x)$ has d roots; $r_1, \ldots, r_d$ then

Only d roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0:
$P(x) = (x-a)Q(x)$.

Proof: $P(x) = (x-a)Q(x) + r$.
Plugin a: $P(a) = r = 0$. $\square$

Lemma 2: $P(x)$ has d roots; $r_1, \ldots, r_d$ then
$P(x) = (x-r_1)(x-r_2)\cdots(x-r_d)c(x)$.

Only d roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0:
$P(x) = (x-a)Q(x)$.

Proof: $P(x) = (x-a)Q(x) + r$.
Plugin a: $P(a) = r = 0$. □

Lemma 2: $P(x)$ has d roots; $r_1, \ldots, r_d$ then
$P(x) = (x-r_1)(x-r_2)\cdots(x-r_d)c(x)$.

Proof Sketch: By induction.

Only *d* roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0:
$P(x) = (x-a)Q(x)$.

Proof: $P(x) = (x-a)Q(x) + r$.
Plugin a: $P(a) = r = 0$. □

Lemma 2: $P(x)$ has d roots; $r_1, \ldots, r_d$ then
$P(x) = (x-r_1)(x-r_2)\cdots(x-r_d)c(x)$.

Proof Sketch: By induction.

Base Case: degree 0.

Only d roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0: $P(x) = (x-a)Q(x)$.

Proof: $P(x) = (x-a)Q(x) + r$.
Plugin a: $P(a) = r = 0$. □

Lemma 2: $P(x)$ has d roots; $r_1, \ldots, r_d$ then
$P(x) = (x-r_1)(x-r_2)\cdots(x-r_d)c(x)$.

Proof Sketch: By induction.

Base Case: degree 0. No roots.

Only d roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0: $P(x) = (x-a)Q(x)$.

Proof: $P(x) = (x-a)Q(x) + r$.
Plugin a: $P(a) = r = 0$. □

Lemma 2: $P(x)$ has d roots; $r_1, \ldots, r_d$ then
$P(x) = (x-r_1)(x-r_2)\cdots(x-r_d)c(x)$.

Proof Sketch: By induction.

Base Case: degree 0. No roots.

Induction Step: $P(x) = (x-r_1)Q(x)$ by Lemma 1.

Only d roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0: $P(x) = (x-a)Q(x)$.

Proof: $P(x) = (x-a)Q(x) + r$.
Plugin a: $P(a) = r = 0$. □

Lemma 2: $P(x)$ has d roots; $r_1, \ldots, r_d$ then
$$P(x) = (x-r_1)(x-r_2)\cdots(x-r_d)c(x).$$

Proof Sketch: By induction.

Base Case: degree 0. No roots.

Induction Step: $P(x) = (x-r_1)Q(x)$ by Lemma 1.
$Q(x)$ has smaller degree ...

Only d roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0:
$P(x) = (x-a)Q(x)$.

Proof: $P(x) = (x-a)Q(x) + r$.
Plugin a: $P(a) = r = 0$. $\square$

Lemma 2: $P(x)$ has d roots; $r_1, \ldots, r_d$ then
$P(x) = (x-r_1)(x-r_2)\cdots(x-r_d)c(x)$.

Proof Sketch: By induction.

Base Case: degree 0. No roots.

Induction Step: $P(x) = (x-r_1)Q(x)$ by Lemma 1.
$Q(x)$ has smaller degree ...
so by induction hypothesis...

Only d roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0:
$P(x) = (x-a)Q(x)$.

Proof: $P(x) = (x-a)Q(x) + r$.
Plugin a: $P(a) = r = 0$. $\square$

Lemma 2: $P(x)$ has d roots; $r_1, \ldots, r_d$ then
$P(x) = (x-r_1)(x-r_2)\cdots(x-r_d)c(x)$.

Proof Sketch: By induction.

Base Case: degree 0. No roots.

Induction Step: $P(x) = (x-r_1)Q(x)$ by Lemma 1.
$Q(x)$ has smaller degree ...
so by induction hypothesis...
we are done.

Only d roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0:
$P(x) = (x-a)Q(x)$.

Proof: $P(x) = (x-a)Q(x) + r$.
Plugin a: $P(a) = r = 0$. □

Lemma 2: $P(x)$ has d roots; $r_1, \ldots, r_d$ then
$P(x) = (x-r_1)(x-r_2)\cdots(x-r_d)c(x)$.

Proof Sketch: By induction.

Base Case: degree 0. No roots.

Induction Step: $P(x) = (x-r_1)Q(x)$ by Lemma 1.
$Q(x)$ has smaller degree ...
so by induction hypothesis...
we are done. □

Only d roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0:
$P(x) = (x-a)Q(x)$.

Proof: $P(x) = (x-a)Q(x) + r$.
Plugin a: $P(a) = r = 0$. □

Lemma 2: $P(x)$ has d roots; $r_1, \ldots, r_d$ then
$P(x) = (x-r_1)(x-r_2)\cdots(x-r_d)c(x)$.

Proof Sketch: By induction.

Base Case: degree 0. No roots.

Induction Step: $P(x) = (x-r_1)Q(x)$ by Lemma 1.
$Q(x)$ has smaller degree ...
so by induction hypothesis...
we are done. □

Thus, $d+1$ roots implies degree is at least $d+1$.

Only d roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0: $P(x) = (x-a)Q(x)$.

Proof: $P(x) = (x-a)Q(x) + r$.
Plugin a: $P(a) = r = 0$. □

Lemma 2: $P(x)$ has d roots; $r_1, \ldots, r_d$ then
$P(x) = (x-r_1)(x-r_2)\cdots(x-r_d)c(x)$.

Proof Sketch: By induction.

Base Case: degree 0. No roots.

Induction Step: $P(x) = (x-r_1)Q(x)$ by Lemma 1.
$Q(x)$ has smaller degree ...
so by induction hypothesis...
we are done. □

Thus, $d+1$ roots implies degree is at least $d+1$.

The contrapositive...

Only d roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0:
$P(x) = (x-a)Q(x)$.

Proof: $P(x) = (x-a)Q(x) + r$.
Plugin a: $P(a) = r = 0$. □

Lemma 2: $P(x)$ has d roots; $r_1, \ldots, r_d$ then
$P(x) = (x-r_1)(x-r_2)\cdots(x-r_d)c(x)$.

Proof Sketch: By induction.

Base Case: degree 0. No roots.

Induction Step: $P(x) = (x-r_1)Q(x)$ by Lemma 1.
$Q(x)$ has smaller degree ...
so by induction hypothesis...
we are done. □

Thus, $d+1$ roots implies degree is at least $d+1$.

The contrapositive...

Roots fact: Any degree d polynomial has at most d roots.

Summary.

Modular Arithmetic Fact: Exactly 1 polynomial of degree $\leq d$ with arithmetic modulo prime p contains $d+1$ pts.

Summary.

Modular Arithmetic Fact: Exactly 1 polynomial of degree $\leq d$ with arithmetic modulo prime p contains $d+1$ pts.

Existence:

Summary.

Modular Arithmetic Fact: Exactly 1 polynomial of degree $\leq d$ with arithmetic modulo prime p contains $d+1$ pts.

Existence:

Lagrange Interpolation.

Summary.

Modular Arithmetic Fact: Exactly 1 polynomial of degree $\leq d$ with arithmetic modulo prime p contains $d+1$ pts.

Existence:

Lagrange Interpolation.

Uniqueness:

Summary.

Modular Arithmetic Fact: Exactly 1 polynomial of degree $\leq d$ with arithmetic modulo prime p contains $d+1$ pts.

Existence:
  Lagrange Interpolation.

Uniqueness:
  At most d roots for degree d polynomial.

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.

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..

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.

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 p is a **finite field** denoted by F_p or $GF(p)$.

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 p is a **finite field** denoted by F_p or $GF(p)$.

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 random $a_1, \ldots, a_{k-1}$.

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 random $a_1,\ldots,a_{k-1}$.
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 random $a_1, \ldots, a_{k-1}$.
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 random $a_1, \ldots, a_{k-1}$.
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 random $a_1,\ldots,a_{k-1}$.
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.

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 random $a_1, \ldots, a_{k-1}$.
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 random $a_1,\ldots,a_{k-1}$.
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 random $a_1, \ldots, a_{k-1}$.
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.
Efficiency: ???

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 $s \in \{0, \dots, p-1\}$

1. Choose $a_0 = s$, and random $a_1, \dots, a_{k-1}$.
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.
Efficiency: ???

Efficiency.

Efficiency.

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

Efficiency.

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

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

Efficiency.

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,

Efficiency.

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,

Efficiency.

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

Efficiency.

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

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.

Efficiency.

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.

Efficiency.

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.
Minimal!

Efficiency.

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

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.
Minimal!

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

Efficiency.

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

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.
Minimal!

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

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

Efficiency.

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.
Minimal!

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

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

(Within 1 bit of) any b-bit string possible!

Efficiency.

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

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.
Minimal!

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

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

(Within 1 bit of) any b-bit string possible!

(Within 1 bit of) b-bits are missing: one $P(i)$.

Efficiency.

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

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.
Minimal!

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

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

(Within 1 bit of) any b-bit string possible!

(Within 1 bit of) b-bits are missing: one $P(i)$.

Within 1 of optimal number of bits.

Runtime.

Runtime.

Runtime: polynomial in k, n, and $\log p$.

1. Evaluate degree $n-1$ polynomial $n+k$ times using $\log p$-bit numbers. $O(kn\log^2 p)$.

Runtime.

Runtime: polynomial in k, n, and $\log p$.

1. Evaluate degree $n-1$ polynomial $n+k$ times using $\log p$-bit numbers. $O(kn\log^2 p)$.
2. Reconstruct secret by solving system of n equations using $\log p$-bit arithmetic. $O(n^3 \log^2 p)$.

Runtime.

Runtime: polynomial in k, n, and $\log p$.

1. Evaluate degree $n-1$ polynomial $n+k$ times using $\log p$-bit numbers. $O(kn\log^2 p)$.
2. Reconstruct secret by solving system of n equations using $\log p$-bit arithmetic. $O(n^3\log^2 p)$.
3. Matrix has special form so $O(n\log n\log^2 p)$ reconstruction.

Runtime.

Runtime: polynomial in k, n, and $\log p$.

1. Evaluate degree $n-1$ polynomial $n+k$ times using $\log p$-bit numbers. $O(kn\log^2 p)$.
2. Reconstruct secret by solving system of n equations using $\log p$-bit arithmetic. $O(n^3\log^2 p)$.
3. Matrix has special form so $O(n\log n\log^2 p)$ reconstruction.

Faster versions in practice are almost as efficient.

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.

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..

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.

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 $GF(m)$.

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 $GF(m)$.

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 $(i, P(i) \mod p)$.

Roubustness: 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 $(i, P(i) \mod p)$.

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.

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 $(i, P(i) \mod p)$.

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.

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,

Minimality.

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

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,

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: $P(1) \dots P(n)$.

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.

Runtime.

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 $GF(m)$?

- m^{d+1}: $d+1$ 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.

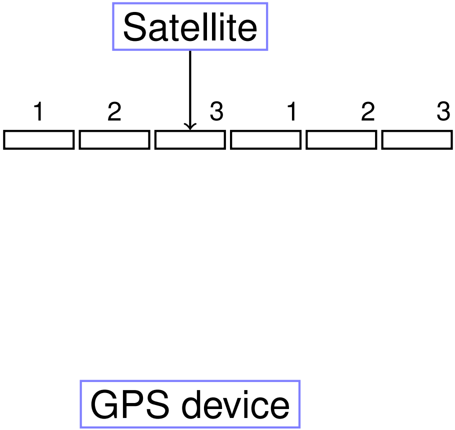
Satellite

3 packet message.

Lose 3 out 6 packets.

GPS device

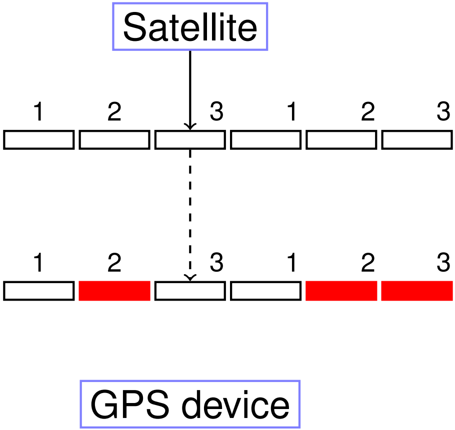
Erasure Codes.



3 packet message. So send 6!

Lose 3 out 6 packets.

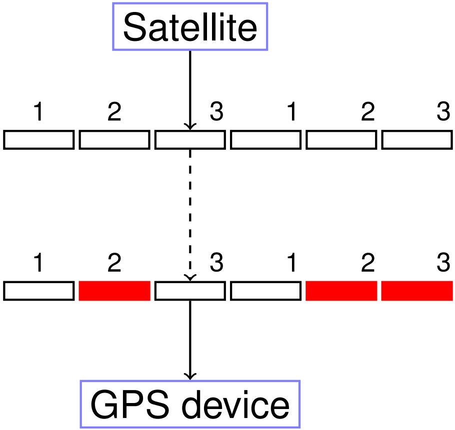
Erasure Codes.



3 packet message. So send 6!

Lose 3 out 6 packets.

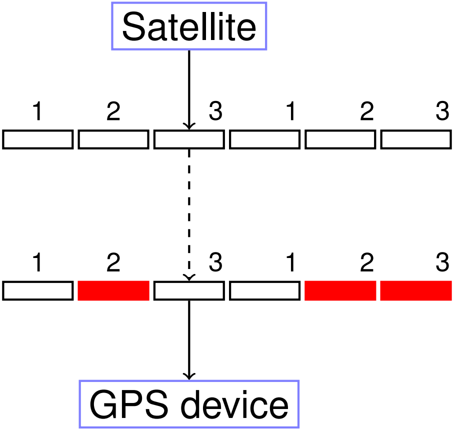
Erasure Codes.



3 packet message. So send 6!

Lose 3 out 6 packets.

Erasure Codes.



3 packet message. So send 6!

Lose 3 out 6 packets.

Gets packets 1,1,and 3.

Solution Idea.

n packet message, channel that loses k packets.

Solution Idea.

n packet message, channel that loses k packets.

Must send $n+k$ packets!

Solution Idea.

n packet message, channel that loses k packets.

Must send $n+k$ packets!

Any n packets

Solution Idea.

n packet message, channel that loses k packets.

Must send $n+k$ packets!

Any n packets should allow reconstruction of n packet message.

Solution Idea.

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.

The Scheme

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

n packet message.

GPS device

Erasure Codes.

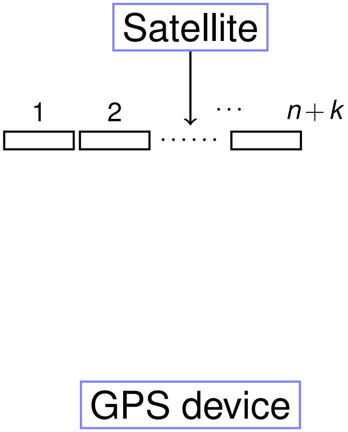
Satellite

n packet message.

Lose *k* packets.

GPS device

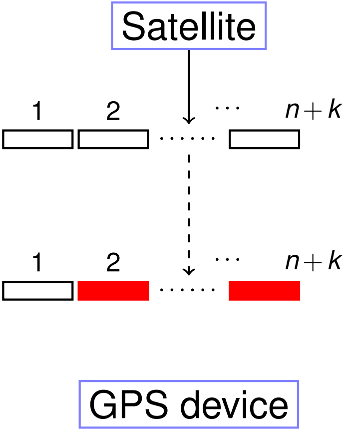
Erasure Codes.



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

Lose k packets.

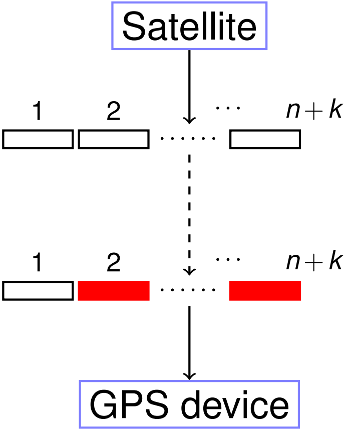
Erasure Codes.



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

Lose k packets.

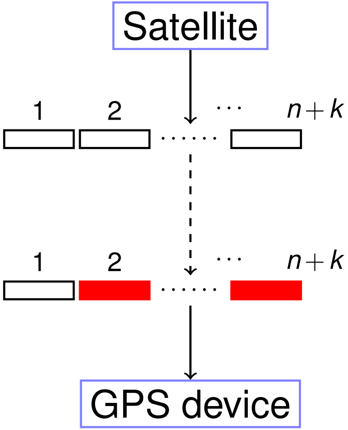
Erasure Codes.



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

Lose k packets.

Erasure Codes.

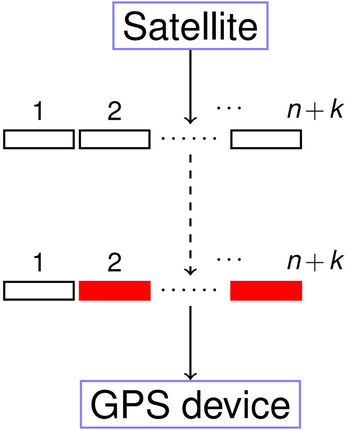


n packet message. So send $n+k$!

Lose k packets.

Any n packets is enough!

Erasure Codes.



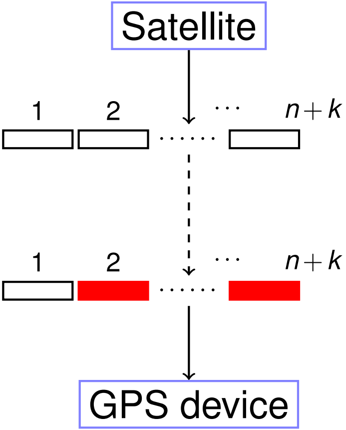
n packet message. So send $n+k$!

Lose k packets.

Any n packets is enough!

n packet message.

Erasure Codes.



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

Lose k packets.

Any n packets is enough!

n packet message.

Optimal.

Information Theory.

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

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.

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.

Information Theory.

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

But: packets need label for x value.

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

– Can also run the Fast Fourier Transform.

Information Theory.

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

But: packets need label for x value.

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

– Can also run the Fast Fourier Transform.

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

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.

Erasure Code: Example.

Send message of 1,4, and 4.

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

How?

Lagrange Interpolation.
Linear System.

Erasure Code: Example.

Send message of 1,4, and 4.

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

How?

Lagrange Interpolation.
Linear System.

Work modulo 5.

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.

$P(x) = x^2 \pmod 5$
$P(1) = 1, P(2) = 4, P(3) = 9 = 4 \pmod 5$

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

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:

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:

$$
\begin{aligned}
P(1) = a_2 + a_1 + a_0 &\equiv 1 \pmod{7} \\
P(2) = 4a_2 + 2a_1 + a_0 &\equiv 4 \pmod{7}
\end{aligned}
$$

Example

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

Modulo 7 to accommodate at least 6 packets.

Linear equations:

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

Example

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

Modulo 7 to accommodate at least 6 packets.

Linear equations:

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

Example

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

Modulo 7 to accommodate at least 6 packets.

Linear equations:

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

$6a_1 + 3a_0 = 2 \pmod 7$,

Example

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

Modulo 7 to accommodate at least 6 packets.

Linear equations:

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

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

Example

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

Modulo 7 to accommodate at least 6 packets.

Linear equations:

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

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

$a_1 = 2a_0$.

Example

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

Modulo 7 to accommodate at least 6 packets.

Linear equations:

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

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

$a_1 = 2a_0$. $a_0 = 2 \pmod{7}$

Example

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

Modulo 7 to accommodate at least 6 packets.

Linear equations:

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

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

$a_1 = 2a_0$. $a_0 = 2 \pmod 7$ $a_1 = 4 \pmod 7$

Example

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

Modulo 7 to accommodate at least 6 packets.

Linear equations:

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

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

$a_1 = 2a_0$. $a_0 = 2 \pmod 7$ $a_1 = 4 \pmod 7$ $a_2 = 2 \pmod 7$

Example

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

Modulo 7 to accommodate at least 6 packets.

Linear equations:

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

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

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

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

Example

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

Modulo 7 to accommodate at least 6 packets.

Linear equations:

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

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

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

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

Example

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

Modulo 7 to accommodate at least 6 packets.

Linear equations:

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

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

$a_1 = 2a_0$. $a_0 = 2 \pmod 7$ $a_1 = 4 \pmod 7$ $a_2 = 2 \pmod 7$

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

$P(1) = 1$,

Example

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

Modulo 7 to accommodate at least 6 packets.

Linear equations:

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

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

$a_1 = 2a_0$. $a_0 = 2 \pmod 7$ $a_1 = 4 \pmod 7$ $a_2 = 2 \pmod 7$

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

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

Example

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

Modulo 7 to accommodate at least 6 packets.

Linear equations:

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

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

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

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

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

Example

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

Modulo 7 to accommodate at least 6 packets.

Linear equations:

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

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

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

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

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

Example

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

Modulo 7 to accommodate at least 6 packets.

Linear equations:

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

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

$a_1 = 2a_0$. $a_0 = 2 \pmod 7$ $a_1 = 4 \pmod 7$ $a_2 = 2 \pmod 7$

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

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

Send

Example

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

Modulo 7 to accommodate at least 6 packets.

Linear equations:

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

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

$a_1 = 2a_0$. $a_0 = 2 \pmod 7$ $a_1 = 4 \pmod 7$ $a_2 = 2 \pmod 7$

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

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

Send
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:

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

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

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

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

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

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

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

Bad reception!

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

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

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

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.

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

Bad reception!

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

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

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

Lagrange or linear equations.

$$\begin{aligned} P(1) = a_2 + a_1 + a_0 &\equiv 1 \pmod 7 \\ P(2) = 4a_2 + 2a_1 + a_0 &\equiv 4 \pmod 7 \\ P(6) = 2a_2 + 3a_1 + a_0 &\equiv 0 \pmod 7 \end{aligned}$$

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.

$$\begin{aligned} P(1) = a_2 + a_1 + a_0 &\equiv 1 \pmod 7 \\ P(2) = 4a_2 + 2a_1 + a_0 &\equiv 4 \pmod 7 \\ P(6) = 2a_2 + 3a_1 + a_0 &\equiv 0 \pmod 7 \end{aligned}$$

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.

$$\begin{aligned} P(1) = a_2 + a_1 + a_0 &\equiv 1 \pmod 7 \\ P(2) = 4a_2 + 2a_1 + a_0 &\equiv 4 \pmod 7 \\ P(6) = 2a_2 + 3a_1 + a_0 &\equiv 0 \pmod 7 \end{aligned}$$

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.

$$\begin{aligned} P(1) = a_2 + a_1 + a_0 &\equiv 1 \pmod 7 \\ P(2) = 4a_2 + 2a_1 + a_0 &\equiv 4 \pmod 7 \\ P(6) = 2a_2 + 3a_1 + a_0 &\equiv 0 \pmod 7 \end{aligned}$$

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.

$$\begin{aligned} P(1) = a_2 + a_1 + a_0 &\equiv 1 \pmod 7 \\ P(2) = 4a_2 + 2a_1 + a_0 &\equiv 4 \pmod 7 \\ P(6) = 2a_2 + 3a_1 + a_0 &\equiv 0 \pmod 7 \end{aligned}$$

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.

$$\begin{aligned} P(1) = a_2 + a_1 + a_0 &\equiv 1 \pmod 7 \\ P(2) = 4a_2 + 2a_1 + a_0 &\equiv 4 \pmod 7 \\ P(6) = 2a_2 + 3a_1 + a_0 &\equiv 0 \pmod 7 \end{aligned}$$

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.

$$\begin{aligned} P(1) = a_2 + a_1 + a_0 &\equiv 1 \pmod 7 \\ P(2) = 4a_2 + 2a_1 + a_0 &\equiv 4 \pmod 7 \\ P(6) = 2a_2 + 3a_1 + a_0 &\equiv 0 \pmod 7 \end{aligned}$$

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.

$$\begin{aligned} P(1) = a_2 + a_1 + a_0 &\equiv 1 \pmod 7 \\ P(2) = 4a_2 + 2a_1 + a_0 &\equiv 4 \pmod 7 \\ P(6) = 2a_2 + 3a_1 + a_0 &\equiv 0 \pmod 7 \end{aligned}$$

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.

$$\begin{aligned} P(1) = a_2 + a_1 + a_0 &\equiv 1 \pmod 7 \\ P(2) = 4a_2 + 2a_1 + a_0 &\equiv 4 \pmod 7 \\ P(6) = 2a_2 + 3a_1 + a_0 &\equiv 0 \pmod 7 \end{aligned}$$

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.

$$\begin{aligned} P(1) = a_2 + a_1 + a_0 &\equiv 1 \pmod 7 \\ P(2) = 4a_2 + 2a_1 + a_0 &\equiv 4 \pmod 7 \\ P(6) = 2a_2 + 3a_1 + a_0 &\equiv 0 \pmod 7 \end{aligned}$$

Channeling Sahai ...

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

Message? $P(1) = 1, P(2) = 4, P(3) = 4$.

Questions for Review

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

Questions for Review

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

How big should modulus be?

Questions for Review

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

- How big should modulus be?
  - Larger than 144

Questions for Review

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

How big should modulus be?

Larger than 144 and prime!

Questions for Review

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

How big should modulus be?

Larger than 144 and prime!

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

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

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.

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?

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

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!

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.

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

Polynomials.

Polynomials.

- ..give Secret Sharing.

Polynomials.

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

Polynomials.

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

Error Correction:

Polynomials.

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

Error Correction:

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

Polynomials.

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

Error Correction:

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

Additional Challenge: Finding which packets are corrupt.

Error Correction

Satellite

GPS device

Error Correction

Satellite

3 packet message.

GPS device

Error Correction

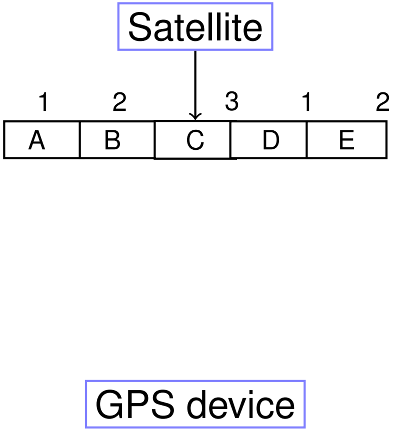
Satellite

3 packet message.

Corrupts 1 packets.

GPS device

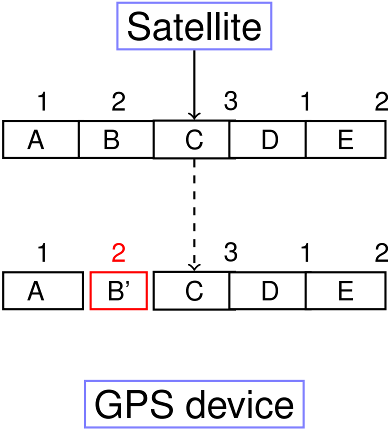
Error Correction



3 packet message. Send 5.

Corrupts 1 packets.

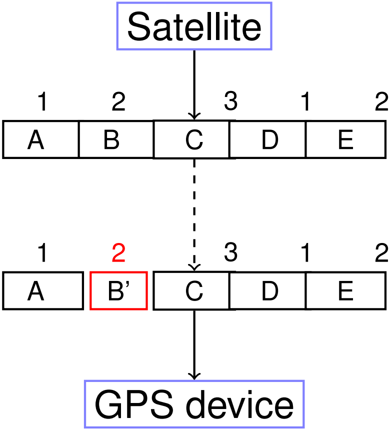
Error Correction



3 packet message. Send 5.

Corrupts 1 packets.

Error Correction



3 packet message. Send 5.

Corrupts 1 packets.

The Scheme.

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

The Scheme.

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

Reed-Solomon Code:

The Scheme.

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

Reed-Solomon Code:

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

The Scheme.

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

Reed-Solomon Code:

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

The Scheme.

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

Reed-Solomon Code:

1. Make a polynomial, $P(x)$ of degree $n-1$,
   that encodes message.
   - $P(1) = m_1, \ldots, P(n) = m_n$.
   - Comment: could encode with packets as coefficients.
2. Send $P(1), \ldots, P(n+2k)$.

The Scheme.

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

Reed-Solomon Code:

1. Make a polynomial, $P(x)$ of degree $n-1$,
   that encodes message.
   - $P(1) = m_1, \ldots, P(n) = m_n$.
   - Comment: could encode with packets as coefficients.
2. Send $P(1), \ldots, P(n+2k)$.

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

The Scheme.

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

Reed-Solomon Code:

1. Make a polynomial, $P(x)$ of degree $n-1$,
   that encodes message.
   - $P(1)=m_1,\ldots,P(n)=m_n$.
   - Comment: could encode with packets as coefficients.
2. Send $P(1),\ldots,P(n+2k)$.

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

Properties:
(1) $P(i)=R(i)$ for at least $n+k$ points i,

The Scheme.

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

Reed-Solomon Code:

1. Make a polynomial, $P(x)$ of degree $n-1$,
   that encodes message.
   - $P(1) = m_1, \ldots, P(n) = m_n$.
   - Comment: could encode with packets as coefficients.
2. Send $P(1), \ldots, P(n+2k)$.

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

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

The Scheme.

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

Reed-Solomon Code:

1. Make a polynomial, $P(x)$ of degree $n-1$,
   that encodes message.
   - $P(1) = m_1, \ldots, P(n) = m_n$.
   - Comment: could encode with packets as coefficients.
2. Send $P(1), \ldots, P(n+2k)$.

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

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

Properties: proof.

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

Properties: proof.

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

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

Properties:

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

Properties: proof.

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

Properties:

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

Properties: proof.

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

Properties:

(1) $P(i) = R(i)$ for at least $n+k$ points i,

(2) $P(x)$ is unique degree $n-1$ polynomial
that contains $\geq n+k$ received points.

Proof:

Properties: proof.

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

Properties:

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

Proof:

(1) Sure.

Properties: proof.

$P(x)$: 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.

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

Properties:

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

Proof:

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

Properties: proof.

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

Properties:

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

Proof:

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

Properties: proof.

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

Properties:

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

Proof:

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

Properties: proof.

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

Properties:

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

Proof:

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

Properties: proof.

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

Properties:

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

Proof:

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

Properties: proof.

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

Properties:

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

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

Properties: proof.

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

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

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

Properties: proof.

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

Properties:

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

Proof:

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

Total points contained by both	: $2n+2k$.	P	Pigeons.
Total points to choose from	: $n+2k$.	H	Holes.
Points contained by both	: $\geq n$.	$\geq P-H$	Collisions.

$\implies Q(i) = P(i)$ at n points.

Properties: proof.

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

Properties:

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

Proof:

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

Total points contained by both	: $2n+2k$.	P	Pigeons.
Total points to choose from	: $n+2k$.	H	Holes.
Points contained by both	: $\geq n$.	$\geq P-H$	Collisions.

$\implies Q(i) = P(i)$ at n points.
$\implies Q(x) = P(x)$.

Properties: proof.

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

Properties:

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

Proof:

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

Total points contained by both	: $2n+2k$.	P	Pigeons.
Total points to choose from	: $n+2k$.	H	Holes.
Points contained by both	: $\geq n$.	$\geq P-H$	Collisions.

$\implies Q(i) = P(i)$ at n points.
$\implies Q(x) = P(x)$. □

Example.

Message: $3, 0, 6$.

Example.

Message: $3, 0, 6$.

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

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

Example.

Message: $3, 0, 6$.

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

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

(Aside: Message in plain text!)

Example.

Message: $3, 0, 6$.

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

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

(Aside: Message in plain text!)

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

Example.

Message: $3, 0, 6$.

Reed Solomon Code: $P(x) = x^2 + x + 1 \pmod 7$ 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.

Slow solution.

Brute Force:

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

Slow solution.

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

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

Slow solution.

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

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

Slow solution.

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

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

Slow solution.

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

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

Slow solution.

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

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

Slow solution.

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

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

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

Example.

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

Example.

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

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

All equations..

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

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..

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

Assume point 1 is wrong

Example.

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

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

All equations..

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

Assume point 1 is wrong and solve..

Example.

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

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

All equations..

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

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..

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

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..

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

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

All equations..

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

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

In general..

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

In general..

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

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

In general..

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

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

In general..

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

$$
\begin{aligned}
p_{n-1} + \cdots p_0 &\equiv R(1) \pmod{p} \\
p_{n-1}2^{n-1} + \cdots p_0 &\equiv R(2) \pmod{p} \\
&\cdot \\
p_{n-1}i^{n-1} + \cdots p_0 &\equiv R(i) \pmod{p} \\
&\cdot \\
p_{n-1}(m)^{n-1} + \cdots p_0 &\equiv R(m) \pmod{p}
\end{aligned}
$$

In general..

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

$$\begin{aligned}
p_{n-1} + \cdots p_0 &\equiv R(1) \pmod{p} \\
p_{n-1}2^{n-1} + \cdots p_0 &\equiv R(2) \pmod{p} \\
&\cdot \\
p_{n-1}i^{n-1} + \cdots p_0 &\equiv R(i) \pmod{p} \\
&\cdot \\
p_{n-1}(m)^{n-1} + \cdots p_0 &\equiv R(m) \pmod{p}
\end{aligned}$$

Error!!

In general..

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

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

Error!! .... Where???

In general..

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

$$\begin{aligned}
p_{n-1} + \cdots p_0 &\equiv R(1) \pmod{p} \\
p_{n-1}2^{n-1} + \cdots p_0 &\equiv R(2) \pmod{p} \\
&\cdot \\
p_{n-1}i^{n-1} + \cdots p_0 &\equiv R(i) \pmod{p} \\
&\cdot \\
p_{n-1}(m)^{n-1} + \cdots p_0 &\equiv R(m) \pmod{p}
\end{aligned}$$

Error!! .... Where???
Could be anywhere!!!

In general..

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

$$
\begin{aligned}
p_{n-1} + \cdots p_0 &\equiv R(1) \pmod p \\
p_{n-1}2^{n-1} + \cdots p_0 &\equiv R(2) \pmod p \\
&\cdot \\
p_{n-1}i^{n-1} + \cdots p_0 &\equiv R(i) \pmod p \\
&\cdot \\
p_{n-1}(m)^{n-1} + \cdots p_0 &\equiv R(m) \pmod p
\end{aligned}
$$

Error!! .... Where???

Could be anywhere!!! ...so try everywhere.

In general..

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

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

Error!! .... Where???
Could be anywhere!!! ...so try everywhere.
Runtime: $\binom{n+2k}{k}$ possibilitities.

In general..

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

$$
\begin{aligned}
p_{n-1} + \cdots p_0 &\equiv R(1) \pmod{p} \\
p_{n-1}2^{n-1} + \cdots p_0 &\equiv R(2) \pmod{p} \\
&. \\
p_{n-1}i^{n-1} + \cdots p_0 &\equiv R(i) \pmod{p} \\
&. \\
p_{n-1}(m)^{n-1} + \cdots p_0 &\equiv R(m) \pmod{p}
\end{aligned}
$$

Error!! .... Where???
Could be anywhere!!! ...so try everywhere.
Runtime: $\binom{n+2k}{k}$ possibilitities.

Something like $(n/k)^k$ ...Exponential in k!.

In general..

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

$$
\begin{aligned}
p_{n-1} + \cdots p_0 &\equiv R(1) \pmod{p} \\
p_{n-1}2^{n-1} + \cdots p_0 &\equiv R(2) \pmod{p} \\
&\cdot \\
p_{n-1}i^{n-1} + \cdots p_0 &\equiv R(i) \pmod{p} \\
&\cdot \\
p_{n-1}(m)^{n-1} + \cdots p_0 &\equiv R(m) \pmod{p}
\end{aligned}
$$

Error!! .... Where???
Could be anywhere!!! ...so try everywhere.
Runtime: $\binom{n+2k}{k}$ possibilitities.

Something like $(n/k)^k$ ...Exponential in k!.

How do we find where the bad packets are efficiently?!?!?!

Ditty...

Oh where, Oh where

Ditty...

Oh where, Oh where
has my little dog gone?

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

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

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

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..

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?

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.

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

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

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
Where, oh where do we look?

Where oh where can my bad packets be?

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

Where oh where can my bad packets be?

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) &\equiv R(2)E(2) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv R(n+2k)E(m) \pmod{p}
\end{aligned}
$$

Where oh where can my bad packets be?

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) &\equiv R(2)E(2) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv R(n+2k)E(m) \pmod{p}
\end{aligned}
$$

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

Where oh where can my bad packets be?

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) &\equiv R(2)E(2) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv R(n+2k)E(m) \pmod{p}
\end{aligned}
$$

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?

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) &\equiv R(2)E(2) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv R(n+2k)E(m) \pmod{p}
\end{aligned}
$$

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?

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) &\equiv R(2)E(2) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv R(n+2k)E(m) \pmod{p}
\end{aligned}
$$

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?

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) &\equiv R(2)E(2) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv R(n+2k)E(m) \pmod{p}
\end{aligned}
$$

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?

$$\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) &\equiv R(2)E(2) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv R(n+2k)E(m) \pmod{p}
\end{aligned}$$

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?

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) &\equiv R(2)E(2) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv R(n+2k)E(m) \pmod{p}
\end{aligned}
$$

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?

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) &\equiv R(2)E(2) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv R(n+2k)E(m) \pmod{p}
\end{aligned}
$$

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?

$$\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) &\equiv R(2)E(2) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv R(n+2k)E(m) \pmod{p}
\end{aligned}$$

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?

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) &\equiv R(2)E(2) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv R(n+2k)E(m) \pmod{p}
\end{aligned}
$$

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?

$$\begin{aligned} E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\ E(2)(p_{n-1}2^{n-1} + \cdots p_0) &\equiv R(2)E(2) \pmod{p} \\ &\vdots \\ E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv R(n+2k)E(m) \pmod{p} \end{aligned}$$

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?

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) &\equiv R(2)E(2) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv R(n+2k)E(m) \pmod{p}
\end{aligned}
$$

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?

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) &\equiv R(2)E(2) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv R(n+2k)E(m) \pmod{p}
\end{aligned}
$$

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?

$$\begin{aligned} E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod p \\ E(2)(p_{n-1}2^{n-1} + \cdots p_0) &\equiv R(2)E(2) \pmod p \\ &\vdots \\ E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv R(n+2k)E(m) \pmod p \end{aligned}$$

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?

$$\begin{aligned} E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\ E(2)(p_{n-1}2^{n-1} + \cdots p_0) &\equiv R(2)E(2) \pmod{p} \\ &\vdots \\ E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv R(n+2k)E(m) \pmod{p} \end{aligned}$$

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?

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod p \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) &\equiv R(2)E(2) \pmod p \\
&\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv R(n+2k)E(m) \pmod p
\end{aligned}
$$

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

Where oh where can my bad packets be?

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) &\equiv R(2)E(2) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv R(n+2k)E(m) \pmod{p}
\end{aligned}
$$

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?

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) &\equiv R(2)E(2) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv R(n+2k)E(m) \pmod{p}
\end{aligned}
$$

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?

$$\begin{aligned} E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\ E(2)(p_{n-1}2^{n-1} + \cdots p_0) &\equiv R(2)E(2) \pmod{p} \\ &\vdots \\ E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv R(n+2k)E(m) \pmod{p} \end{aligned}$$

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).)

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

Plugin points...

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

Example.

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

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

Plugin points...

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

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

Example.

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

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

Plugin points...

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

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

Plugin points...

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

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...

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

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

Plugin points...

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

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

Plugin points...

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

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

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

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

Example.

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

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

Plugin points...

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

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

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

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

Example.

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

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

Plugin points...

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

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...

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

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,

$$
\begin{aligned}
(p_{n-1} + \cdots p_0) &\equiv R(1) \pmod{p} \\
&\vdots \\
(p_{n-1} i^{n-1} + \cdots p_0) &\equiv R(i) \pmod{p} \\
&\vdots \\
(p_{n-1} (n+2k)^{n-1} + \cdots p_0) &\equiv R(m) \pmod{p}
\end{aligned}
$$

..turn their heads each day,

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
&\vdots \\
E(i)(p_{n-1} i^{n-1} + \cdots p_0) &\equiv R(i)E(i) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(n+2k)^{n-1} + \cdots p_0) &\equiv R(m)E(m) \pmod{p}
\end{aligned}
$$

...so satisfied, I'm on my way.

..turn their heads each day,

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
&\vdots \\
E(i)(p_{n-1} i^{n-1} + \cdots p_0) &\equiv R(i)E(i) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(n+2k)^{n-1} + \cdots p_0) &\equiv R(m)E(m) \pmod{p}
\end{aligned}
$$

...so satisfied, I'm on my way.

$m = n + 2k$ satisfied equations,

..turn their heads each day,

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod p \\
&\vdots \\
E(i)(p_{n-1}i^{n-1} + \cdots p_0) &\equiv R(i)E(i) \pmod p \\
&\vdots \\
E(m)(p_{n-1}(n+2k)^{n-1} + \cdots p_0) &\equiv R(m)E(m) \pmod p
\end{aligned}
$$

...so satisfied, I'm on my way.

$m = n + 2k$ satisfied equations, $n + k$ unknowns.

..turn their heads each day,

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
&\vdots \\
E(i)(p_{n-1} i^{n-1} + \cdots p_0) &\equiv R(i)E(i) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(n+2k)^{n-1} + \cdots p_0) &\equiv R(m)E(m) \pmod{p}
\end{aligned}
$$

...so satisfied, I'm on my way.

$m = n+2k$ satisfied equations, $n+k$ unknowns. But nonlinear!

..turn their heads each day,

$$\begin{aligned} E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\ &\vdots \\ E(i)(p_{n-1} i^{n-1} + \cdots p_0) &\equiv R(i)E(i) \pmod{p} \\ &\vdots \\ E(m)(p_{n-1}(n+2k)^{n-1} + \cdots p_0) &\equiv R(m)E(m) \pmod{p} \end{aligned}$$

...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,

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
&\vdots \\
E(i)(p_{n-1} i^{n-1} + \cdots p_0) &\equiv R(i)E(i) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(n+2k)^{n-1} + \cdots p_0) &\equiv R(m)E(m) \pmod{p}
\end{aligned}
$$

...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,

$$\begin{aligned} E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\ &\vdots \\ E(i)(p_{n-1}i^{n-1} + \cdots p_0) &\equiv R(i)E(i) \pmod{p} \\ &\vdots \\ E(m)(p_{n-1}(n+2k)^{n-1} + \cdots p_0) &\equiv R(m)E(m) \pmod{p} \end{aligned}$$

...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,

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
&\vdots \\
E(i)(p_{n-1} i^{n-1} + \cdots p_0) &\equiv R(i)E(i) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(n+2k)^{n-1} + \cdots p_0) &\equiv R(m)E(m) \pmod{p}
\end{aligned}
$$

...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,

$$
\begin{aligned}
E(1)(p_{n-1} + \cdots p_0) &\equiv R(1)E(1) \pmod{p} \\
&\vdots \\
E(i)(p_{n-1} i^{n-1} + \cdots p_0) &\equiv R(i)E(i) \pmod{p} \\
&\vdots \\
E(m)(p_{n-1}(n+2k)^{n-1} + \cdots p_0) &\equiv R(m)E(m) \pmod{p}
\end{aligned}
$$

...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 ...

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

$\Longrightarrow$ k (unknown) coefficients.

Finding $Q(x)$ and $E(x)$?

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

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

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

Finding $Q(x)$ and $E(x)$?

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

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

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

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

Finding $Q(x)$ and $E(x)$?

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

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

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

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

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

Finding $Q(x)$ and $E(x)$?

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

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

$\implies$ k (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 ...

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

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

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

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

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

Number of unknown coefficients:

Finding $Q(x)$ and $E(x)$?

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

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

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

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

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

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

Number of unknown coefficients: $n+2k$.

Solving for $Q(x)$ and $E(x)$...

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

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

Solving for $Q(x)$ and $E(x)$...

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

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

Gives $n+2k$ linear equations.

Solving for $Q(x)$ and $E(x)$...

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

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

Gives $n+2k$ linear equations.

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

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.

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

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.

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

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.

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

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

Solving for $Q(x)$ and $E(x)$...

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

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

Gives $n+2k$ linear equations.

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

..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 $1,\dots,i,n+2k=m$,

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

Gives $n+2k$ linear equations.

$$\begin{aligned}
a_{n+k-1} + \dots a_0 &\equiv R(1)(1 + b_{k-1}\cdots b_0) \pmod{p} \\
a_{n+k-1}(2)^{n+k-1} + \dots a_0 &\equiv R(2)((2)^k + b_{k-1}(2)^{k-1}\cdots b_0) \pmod{p} \\
&\vdots \\
a_{n+k-1}(m)^{n+k-1} + \dots a_0 &\equiv R(m)((m)^k + b_{k-1}(m)^{k-1}\cdots b_0) \pmod{p}
\end{aligned}$$

..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.

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

..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.

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

..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.

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

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

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

Find $P(x) = Q(x)/E(x)$.

Example.

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

Example.

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

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

Example.

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

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

$E(x) = x - b_0$

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

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

$E(x) = x - b_0$

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

$$a_3 + a_2 + a_1 + a_0 \equiv 3(1 - b_0) \pmod 7$$

Example.

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

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

$E(x) = x - b_0$

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

$$\begin{aligned} a_3 + a_2 + a_1 + a_0 &\equiv 3(1 - b_0) \pmod 7 \\ a_3 + 4a_2 + 2a_1 + a_0 &\equiv 1(2 - b_0) \pmod 7 \end{aligned}$$

Example.

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

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

$E(x) = x - b_0$

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

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

Example.

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

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

$E(x) = x - b_0$

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

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

$a_3 = 1$, $a_2 = 6$, $a_1 = 6$, $a_0 = 5$ and $b_0 = 2$.

Example.

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

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

$E(x) = x - b_0$

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

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

$a_3 = 1$, $a_2 = 6$, $a_1 = 6$, $a_0 = 5$ and $b_0 = 2$.

$Q(x) = x^3 + 6x^2 + 6x + 5.$

Example.

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

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

$E(x) = x - b_0$

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

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

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

$Q(x) = x^3 + 6x^2 + 6x + 5.$
$E(x) = x - 2.$

```
           -----------------
x - 2 ) x^3  + 6 x^2 + 6 x + 5
```

Example: finishing up.

$Q(x) = x^3 + 6x^2 + 6x + 5.$
$E(x) = x - 2.$

```
                   1 x^2
              -----------------
x - 2 )  x^3   + 6 x^2 + 6 x + 5
         x^3   - 2 x^2
```

Example: finishing up.

$Q(x) = x^3 + 6x^2 + 6x + 5.$
$E(x) = x - 2.$

```
                 1 x^2
          -----------------
x - 2 ) x^3  + 6 x^2 + 6 x + 5
        x^3  - 2 x^2
        ----------
                 1 x^2 + 6 x + 5
```

Example: finishing up.

$Q(x) = x^3 + 6x^2 + 6x + 5.$
$E(x) = x - 2.$

```
                1 x^2 + 1 x
         -----------------
x - 2 )  x^3   + 6 x^2 + 6 x + 5
         x^3   - 2 x^2
         ----------
                1 x^2 + 6 x + 5
                1 x^2 - 2 x
```

Example: finishing up.

$Q(x) = x^3 + 6x^2 + 6x + 5.$
$E(x) = x - 2.$

```
                1 x^2 + 1 x
         -----------------
x - 2 )  x^3  + 6 x^2 + 6 x + 5
         x^3  - 2 x^2
         ----------
                1 x^2 + 6 x + 5
                1 x^2 - 2 x
                ---------------
                          x + 5
```

Example: finishing up.

$Q(x) = x^3 + 6x^2 + 6x + 5.$
$E(x) = x - 2.$

```
                   1 x^2 + 1 x + 1
          -----------------
x - 2 ) x^3  + 6 x^2 + 6 x + 5
        x^3  - 2 x^2
        ----------
                   1 x^2 + 6 x + 5
                   1 x^2 - 2 x
                   ---------------
                             x + 5
                             x - 2
```

Example: finishing up.

$Q(x) = x^3 + 6x^2 + 6x + 5.$
$E(x) = x - 2.$

```
                   1 x^2 + 1 x + 1
          -----------------
x - 2 )  x^3  + 6 x^2 + 6 x + 5
         x^3  - 2 x^2
         ----------
                 1 x^2 + 6 x + 5
                 1 x^2 - 2 x
                 ---------------
                           x + 5
                           x - 2
                           -----
                               0
```

Example: finishing up.

$Q(x) = x^3 + 6x^2 + 6x + 5.$
$E(x) = x - 2.$

```
                1 x^2 + 1 x + 1
            -----------------
    x - 2 ) x^3  + 6 x^2 + 6 x + 5
            x^3  - 2 x^2
            ----------
                1 x^2 + 6 x + 5
                1 x^2 - 2 x
                ---------------
                          x + 5
                          x - 2
                          -----
                              0
```

$P(x) = x^2 + x + 1$

Example: finishing up.

$Q(x) = x^3 + 6x^2 + 6x + 5.$
$E(x) = x - 2.$

```
                  1 x^2 + 1 x + 1
         -----------------
x - 2 )  x^3  + 6 x^2 + 6 x + 5
         x^3  - 2 x^2
         ----------
                  1 x^2 + 6 x + 5
                  1 x^2 - 2 x
                  ---------------
                            x + 5
                            x - 2
                            -----
                                0
```

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

Example: finishing up.

$Q(x) = x^3 + 6x^2 + 6x + 5.$
$E(x) = x - 2.$

```
                1 x^2 + 1 x + 1
         -----------------
x - 2 )  x^3  + 6 x^2 + 6 x + 5
         x^3  - 2 x^2
         ----------
                1 x^2 + 6 x + 5
                1 x^2 - 2 x
                ---------------
                          x + 5
                          x - 2
                          -----
                              0
```

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

What is $\frac{x-2}{x-2}$?

Example: finishing up.

$Q(x) = x^3 + 6x^2 + 6x + 5.$
$E(x) = x - 2.$

```
                   1 x^2 + 1 x + 1
            -----------------
    x - 2 ) x^3  + 6 x^2 + 6 x + 5
            x^3  - 2 x^2
            ----------
                   1 x^2 + 6 x + 5
                   1 x^2 - 2 x
                   ---------------
                             x + 5
                             x - 2
                             -----
                                 0
```

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

What is $\frac{x-2}{x-2}$? 1

Example: finishing up.

$Q(x) = x^3 + 6x^2 + 6x + 5.$
$E(x) = x - 2.$

```
                 1 x^2 + 1 x + 1
          -----------------
x - 2 )  x^3  + 6 x^2 + 6 x + 5
         x^3  - 2 x^2
         ----------
                 1 x^2 + 6 x + 5
                 1 x^2 - 2 x
                 ---------------
                           x + 5
                           x - 2
                           -----
                               0
```

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

What is $\frac{x-2}{x-2}$? 1
Except at $x = 2$?

Example: finishing up.

$Q(x) = x^3 + 6x^2 + 6x + 5.$
$E(x) = x - 2.$

```
                1 x^2 + 1 x + 1
         -----------------
x - 2 )  x^3  + 6 x^2 + 6 x + 5
         x^3  - 2 x^2
         ----------
                1 x^2 + 6 x + 5
                1 x^2 - 2 x
                ---------------
                          x + 5
                          x - 2
                          -----
                              0
```

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

What is $\frac{x-2}{x-2}$? 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 undersanding.

You have error locator polynomial!

Check your undersanding.

You have error locator polynomial!

Where oh where have my packets gone wrong?

Check your undersanding.

You have error locator polynomial!

Where oh where have my packets gone wrong?

Factor?

Check your undersanding.

You have error locator polynomial!

Where oh where have my packets gone wrong?

Factor? Sure.

Check your undersanding.

You have error locator polynomial!

Where oh where have my packets gone wrong?

Factor? Sure.
Check all values?

Check your undersanding.

You have error locator polynomial!

Where oh where have my packets gone wrong?

Factor? Sure.
Check all values? Sure.

Check your undersanding.

You have error locator polynomial!

Where oh where have my packets gone wrong?

Factor? Sure.
Check all values? Sure.

Check your undersanding.

You have error locator polynomial!

Where oh where have my packets gone wrong?

Factor? Sure.
Check all values? Sure.

Efficiency?

Check your undersanding.

You have error locator polynomial!

Where oh where have my packets gone wrong?

Factor? Sure.
Check all values? Sure.

Efficiency? Sure.

Check your undersanding.

You have error locator polynomial!

Where oh where have my packets gone wrong?

Factor? Sure.
Check all values? Sure.

Efficiency? Sure. Only $n+2k$ values.

Check your undersanding.

You have error locator polynomial!

Where oh where have my packets gone wrong?

Factor? Sure.
Check all values? Sure.

Efficiency? Sure. Only $n+2k$ values.
See where it is 0.

Hmmm...

Is there one and only one $P(x)$ from Berlekamp-Welsh procedure?

Hmmm...

Is there one and only one $P(x)$ from Berlekamp-Welsh procedure?

Existence: there is a $P(x)$ and $E(x)$ that satisfy equations.

Unique solution for $P(x)$

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

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

Unique solution for $P(x)$

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

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

Proof:

Unique solution for $P(x)$

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

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

Proof:
We claim

Unique solution for $P(x)$

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

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

Proof:
We claim

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

Unique solution for $P(x)$

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

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

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:

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 $n+2k-1$

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

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.

Unique solution for $P(x)$

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

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

Proof:
We claim

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

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

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

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

Unique solution for $P(x)$

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

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

Proof:
We claim

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

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.
$\Longrightarrow \frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)}$ equal on n points.
Both degree $\leq n-1$

Unique solution for $P(x)$

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

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

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.
$\implies \frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)}$ equal on n points.
Both degree $\leq n-1 \implies$ Same polynomial!

Unique solution for $P(x)$

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

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

Proof:
We claim

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

Equation 2 implies 1:

$Q'(x)E(x)$ and $Q(x)E'(x)$ are degree $n+2k-1$
and agree on $n+2k$ points
$E(x)$ and $E'(x)$ have at most k zeros each.
Can cross divide at n points.
$\implies \frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)}$ equal on n points.
Both degree $\leq n-1 \implies$ Same polynomial! □

Last bit.

Fact: $Q'(x)E(x) = Q(x)E'(x)$ on $n+2k$ values of x.

Last bit.

Fact: $Q'(x)E(x) = Q(x)E'(x)$ on $n+2k$ values of x.

Proof:

Last bit.

Fact: $Q'(x)E(x) = Q(x)E'(x)$ on $n+2k$ values of x.

Proof: Construction implies that

Last bit.

Fact: $Q'(x)E(x) = Q(x)E'(x)$ on $n+2k$ values of x.

Proof: Construction implies that

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

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

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 $i \in \{1, \ldots n+2k\}$.

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 $Q(i) = 0$.

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 $Q'(i) = 0$.

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 $Q'(i) = 0$.
$\implies Q(i)E'(i) = Q'(i)E(i)$ holds when $E(i)$ or $E'(i)$ are zero.

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 $Q'(i) = 0$.
$\implies Q(i)E'(i) = Q'(i)E(i)$ holds when $E(i)$ or $E'(i)$ are zero.

When $E'(i)$ and $E(i)$ are not zero

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 $Q'(i) = 0$.
$\implies Q(i)E'(i) = Q'(i)E(i)$ holds when $E(i)$ or $E'(i)$ are zero.

When $E'(i)$ and $E(i)$ are not zero

$$\frac{Q'(i)}{E'(i)} = \frac{Q(i)}{E(i)} = R(i).$$

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 $Q'(i) = 0$.
$\implies$ $Q(i)E'(i) = Q'(i)E(i)$ 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.

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 $Q'(i) = 0$.
$\implies Q(i)E'(i) = Q'(i)E(i)$ 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. □

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 $i \in \{1, \dots n+2k\}$.

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!

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 $Q'(i) = 0$.
$\implies Q(i)E'(i) = Q'(i)E(i)$ 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 $x = 2$.

Yaay!!

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 $P(1),...P(8)$.

You recieve 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.

Poll

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

You recieve 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.

Poll

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

You recieve 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 coefficents in $E(x)$ is 2.
(C) The number of unknown coefficents 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 $P(1),...P(8)$.

You recieve 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 coefficents in $E(x)$ is 2.
(C) The number of unknown coefficents 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?

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? $n-1$.

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?

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 $Q(x) = E(x)P(x)$.

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.

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.

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. Welsh-Berlekamp Decoding.

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. Welsh-Berlekamp Decoding. Perfection!

Cool.

Really Cool!