
1. Finish Polynomials and Secrets.
2. Finite Fields: Abstract Algebra
3. Erasure Coding
Modular Arithmetic Fact and Secrets

Modular Arithmetic Fact:
There is exactly 1 polynomial of degree \( \leq d \) with arithmetic modulo prime \( p \) that contains \( d + 1 \) pts.

Note: The points have to have different \( x \) values!

Shamir's \( k \) out of \( n \) Scheme:

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 \( i \) for \( i \geq 1 \) is point \( (i, P(i) \mod p) \).

Robustness: Any \( k \) shares gives secret.
Knowing \( k \) pts, find unique \( P(x) \), evaluate \( P(0) \).

Secrecy: Any \( k - 1 \) shares give nothing.
Knowing \( \leq k - 1 \) pts, any \( P(0) \) is possible.
Modular Arithmetic Fact: There is exactly 1 polynomial of degree \( \leq d \) with arithmetic modulo prime \( p \) that contains \( d + 1 \) pts.
Modular Arithmetic Fact and Secrets

**Modular Arithmetic Fact:** There is exactly 1 polynomial of degree \( \leq d \) with arithmetic modulo prime \( p \) that contains \( d + 1 \) pts.

Note: The points have to have different \( x \) values!
Modular Arithmetic Fact and Secrets

**Modular Arithmetic Fact:** There is exactly 1 polynomial of degree \( \leq d \) with arithmetic modulo prime \( p \) that contains \( d + 1 \) pts.

Note: The points have to have different \( x \) values!

**Shamir’s \( k \) out of \( n \) Scheme:**
Modular Arithmetic Fact and Secrets

**Modular Arithmetic Fact:** There is exactly 1 polynomial of degree \( \leq d \) with arithmetic modulo prime \( p \) that contains \( d + 1 \) pts.

Note: The points have to have different \( x \) values!

**Shamir’s \( k \) out of \( n \) Scheme:**

Secret \( s \in \{0, \ldots, p - 1\} \)
Modular Arithmetic Fact and Secrets

**Modular Arithmetic Fact:** There is exactly 1 polynomial of degree \( \leq d \) with arithmetic modulo prime \( p \) that contains \( d + 1 \) pts.

Note: The points have to have different \( x \) values!

**Shamir’s \( k \) out of \( n \) Scheme:**

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

1. Choose \( a_0 = s \), and random \( a_1, \ldots, a_{k-1} \).
Modular Arithmetic Fact: There is exactly 1 polynomial of degree \( \leq d \) with arithmetic modulo prime \( p \) that contains \( d + 1 \) pts.

Note: The points have to have different \( x \) values!

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 \( a_0 = s \).
Modular Arithmetic Fact: There is exactly 1 polynomial of degree \( \leq d \) with arithmetic modulo prime \( p \) that contains \( d + 1 \) pts.

Note: The points have to have different \( x \) values!

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 \( i \) for \( i \geq 1 \) is point \((i, P(i) \mod p)\).
Modular Arithmetic Fact: There is exactly 1 polynomial of degree \( \leq d \) with arithmetic modulo prime \( p \) that contains \( d + 1 \) pts.

Note: The points have to have different \( x \) values!

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 \( i \) for \( i \geq 1 \) is point \((i, P(i) \mod p)\).

Robustness: Any \( k \) shares gives secret.
Modular Arithmetic Fact and Secrets

**Modular Arithmetic Fact:** There is exactly 1 polynomial of degree \( \leq d \) with arithmetic modulo prime \( p \) that contains \( d + 1 \) pts.

Note: The points have to have different \( x \) values!

**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 \( i \) for \( i \geq 1 \) is point \( (i, P(i) \mod p) \).

**Robustness:** Any \( k \) shares gives secret.
**Modular Arithmetic Fact:** There is exactly 1 polynomial of degree \( \leq d \) with arithmetic modulo prime \( p \) that contains \( d + 1 \) pts.

Note: The points have to have different \( x \) values!

**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 \( i \) for \( i \geq 1 \) is point \((i, P(i) \mod p)\).

**Robustness:** Any \( k \) shares gives secret.
Knowing \( k \) pts, find unique \( P(x) \), evaluate \( P(0) \).
**Modular Arithmetic Fact:** There is exactly 1 polynomial of degree \( \leq d \) with arithmetic modulo prime \( p \) that contains \( d + 1 \) pts.

Note: The points have to have different \( x \) values!

**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 \( i \) for \( i \geq 1 \) is point \( (i, P(i) \mod p) \).

**Robustness:** Any \( k \) shares gives secret.
Knowing \( k \) pts, find unique \( P(x) \), evaluate \( P(0) \).

**Secrecy:** Any \( k-1 \) shares give nothing.
Modular Arithmetic Fact: There is exactly 1 polynomial of degree \( \leq d \) with arithmetic modulo prime \( p \) that contains \( d + 1 \) pts.

Note: The points have to have different \( x \) values!

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 \( i \) for \( i \geq 1 \) is point \( (i, P(i) \mod p) \).

Robustness: Any \( k \) shares gives secret. Knowing \( k \) pts, find unique \( P(x) \), evaluate \( P(0) \).
Secrecy: Any \( k - 1 \) shares give nothing.
Modular Arithmetic Fact: There is exactly 1 polynomial of degree \( \leq d \) with arithmetic modulo prime \( p \) that contains \( d + 1 \) pts.

Note: The points have to have different \( x \) values!

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 \( i \) for \( i \geq 1 \) is point \((i, P(i) \mod p)\).

Robustness: Any \( k \) shares gives secret.
Knowing \( k \) pts, find unique \( P(x) \), evaluate \( P(0) \).

Secrecy: Any \( k - 1 \) shares give nothing.
Knowing \( \leq k - 1 \) pts, any \( P(0) \) is possible.
There exists a polynomial...
There exists a polynomial...

**Modular Arithmetic Fact:** Exactly 1 degree \( \leq d \) polynomial with arithmetic modulo prime \( p \) contains \( d + 1 \) pts.
There exists a polynomial...

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

**Proof of at least one polynomial:**
Given points: \((x_1, y_1); (x_2, y_2) \cdots (x_{d+1}, y_{d+1})\).
There exists a polynomial...

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

**Proof of at least one polynomial:**
Given points: $(x_1, y_1); (x_2, y_2) \cdots (x_{d+1}, y_{d+1})$.

$$
\Delta_i(x) = \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)}.
$$
There exists a polynomial...

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

**Proof of at least one polynomial:**
Given points: \((x_1, y_1); (x_2, y_2) \cdots (x_{d+1}, y_{d+1})\).

\[
\Delta_i(x) = \frac{\prod_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)}.
\]

Numerator is 0 at \( x_j \neq x_i \).
There exists a polynomial...

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

**Proof of at least one polynomial:**
Given points: \((x_1, y_1); (x_2, y_2) \cdots (x_{d+1}, y_{d+1})\).

\[
\Delta_i(x) = \frac{\prod_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)}.
\]

Numerator is 0 at \( x_j \neq x_i \).
Denominator makes it 1 at \( x_i \).
There exists a polynomial...

**Modular Arithmetic Fact:** Exactly 1 degree ≤ d polynomial with arithmetic modulo prime p contains d + 1 pts.

**Proof of at least one polynomial:**

Given points: (x₁, y₁); (x₂, y₂) ··· (x_{d+1}, y_{d+1}).

\[ \Delta_i(x) = \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)}. \]

Numerator is 0 at \( x_j \neq x_i \).

Denominator makes it 1 at \( x_i \).

And .

\[ P(x) = y_1 \Delta_1(x) + y_2 \Delta_2(x) + \cdots + y_{d+1} \Delta_{d+1}(x). \]
There exists a polynomial...

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

**Proof of at least one polynomial:**
Given points: \((x_1, y_1); (x_2, y_2) \cdots (x_{d+1}, y_{d+1})\).

\[
\Delta_i(x) = \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)}.
\]

Numerator is 0 at \( x_j \neq x_i \).

Denominator makes it 1 at \( x_i \).

And..

\[
P(x) = y_1 \Delta_1(x) + y_2 \Delta_2(x) + \cdots + y_{d+1} \Delta_{d+1}(x).
\]

hits points \((x_1, y_1); (x_2, y_2) \cdots (x_{d+1}, y_{d+1})\).
There exists a polynomial...

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

**Proof of at least one polynomial:**
Given points: $(x_1, y_1); (x_2, y_2) \cdots (x_{d+1}, y_{d+1})$.

$$\Delta_i(x) = \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)}.$$ 

Numerator is 0 at $x_j \neq x_i$.
Denominator makes it 1 at $x_i$.
And..

$$P(x) = y_1 \Delta_1(x) + y_2 \Delta_2(x) + \cdots + y_{d+1} \Delta_{d+1}(x).$$

hits points $(x_1, y_1); (x_2, y_2) \cdots (x_{d+1}, y_{d+1})$. Degree $d$ polynomial!
There exists a polynomial...

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

**Proof of at least one polynomial:**
Given points: \((x_1, y_1); (x_2, y_2) \cdots (x_{d+1}, y_{d+1})\).

\[
\Delta_i(x) = \frac{\prod_{j\neq i}(x - x_j)}{\prod_{j\neq i}(x_i - x_j)}.
\]

Numerator is 0 at \( x_j \neq x_i \).
Denominator makes it 1 at \( x_i \).

And..

\[
P(x) = y_1\Delta_1(x) + y_2\Delta_2(x) + \cdots + y_{d+1}\Delta_{d+1}(x).
\]

hits points \((x_1, y_1); (x_2, y_2) \cdots (x_{d+1}, y_{d+1})\). Degree \( d \) polynomial!

Construction proves the existence of a polynomial!
Reiterating Examples.

\[ \Delta_i(x) = \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)}. \]
Reiterating Examples.

\[ \Delta_i(x) = \frac{\prod_{j \neq i} (x-x_j)}{\prod_{j \neq i} (x_i-x_j)}. \]

Degree 1 polynomial, \( P(x) \), that contains \((1,3)\) and \((3,4)\)?

Put the delta functions together.
Reiterating Examples.

\[ \Delta_i(x) = \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)}. \]

Degree 1 polynomial, \( P(x) \), that contains \((1, 3)\) and \((3, 4)\)?

Work modulo 5.
Reiterating Examples.

\[ \Delta_i(x) = \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)}. \]

Degree 1 polynomial, \( P(x) \), that contains (1,3) and (3,4)? Work modulo 5.
\( \Delta_1(x) \) contains (1,1) and (3,0).
Reiterating Examples.

\[ \Delta_i(x) = \frac{\prod_{j \neq i} (x-x_j)}{\prod_{j \neq i} (x_i-x_j)}. \]

Degree 1 polynomial, \( P(x) \), that contains \((1, 3)\) and \((3, 4)\)?

Work modulo 5.

\( \Delta_1(x) \) contains \((1, 1)\) and \((3, 0)\).

\[ \Delta_1(x) = \frac{(x-3)}{1-3} = \frac{x-3}{-2} \]
Reiterating Examples.

\[ \Delta_i(x) = \frac{\prod_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)}. \]

Degree 1 polynomial, \( P(x) \), that contains (1, 3) and (3, 4)?
Work modulo 5.
\( \Delta_1(x) \) contains (1, 1) and (3, 0).

\[ \Delta_1(x) = \frac{(x-3)}{1-3} = \frac{x-3}{-2} = 2(x - 3) \]
Reiterating Examples.

\[ \Delta_i(x) = \frac{\prod_{j \neq i} (x-x_j)}{\prod_{j \neq i} (x_i-x_j)}. \]

Degree 1 polynomial, \( P(x) \), that contains \((1,3)\) and \((3,4)\)?

Work modulo 5.

\( \Delta_1(x) \) contains \((1,1)\) and \((3,0)\).

\[ \Delta_1(x) = \frac{(x-3)}{1-3} = \frac{x-3}{-2} = 2(x - 3) = 2x - 6 \]
Reiterating Examples.

\[ \Delta_i(x) = \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)}. \]

Degree 1 polynomial, \( P(x) \), that contains (1, 3) and (3, 4)? Work modulo 5.

\( \Delta_1(x) \) contains (1, 1) and (3, 0).

\[
\Delta_1(x) = \frac{(x-3)}{1-3} = \frac{x-3}{-2} = 2(x - 3) = 2x - 6 = 2x + 4 \pmod{5}.
\]
Reiterating Examples.

\[ \Delta_i(x) = \frac{\prod_{j \neq i} (x-x_j)}{\prod_{j \neq i} (x_i-x_j)}. \]

Degree 1 polynomial, \( P(x) \), that contains \((1,3)\) and \((3,4)\)?

Work modulo 5.

\[ \Delta_1(x) \] contains \((1,1)\) and \((3,0)\).

\[ \Delta_1(x) = \frac{(x-3)}{1-3} = \frac{x-3}{-2} = 2(x-3) = 2x - 6 = 2x + 4 \pmod{5}. \]

For a quadratic, \( a_2x^2 + a_1x + a_0 \) hits \((1,3); (2,4); (3,0)\).
Reiterating Examples.

\[ \Delta_i(x) = \frac{\prod_{j \neq i}(x-x_j)}{\prod_{j \neq i}(x_i-x_j)}. \]

Degree 1 polynomial, \( P(x) \), that contains \((1,3)\) and \((3,4)\)?
Work modulo 5.
\( \Delta_1(x) \) contains \((1,1)\) and \((3,0)\).

\[
\Delta_1(x) = \frac{(x-3)}{1-3} = \frac{x-3}{-2} = 2(x-3) = 2x - 6 = 2x + 4 \pmod{5}.
\]

For a quadratic, \( a_2 x^2 + a_1 x + a_0 \) hits \((1,3);(2,4);(3,0)\).
Work modulo 5.
Reiterating Examples.

\[ \Delta_i(x) = \frac{\prod_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)}. \]

Degree 1 polynomial, \( P(x) \), that contains \((1, 3)\) and \((3, 4)\)?

Work modulo 5.

\( \Delta_1(x) \) contains \((1, 1)\) and \((3, 0)\).

\[ \Delta_1(x) = \frac{(x - 3)}{1 - 3} = \frac{x - 3}{-2} = 2(x - 3) = 2x - 6 = 2x + 4 \pmod{5}. \]

For a quadratic, \( a_2 x^2 + a_1 x + a_0 \) hits \((1, 3); (2, 4); (3, 0)\).

Work modulo 5.

Find \( \Delta_1(x) \) polynomial contains \((1, 1); (2, 0); (3, 0)\).
Reiterating Examples.

\[ \Delta_i(x) = \frac{\prod_{j \neq i}(x-x_j)}{\prod_{j \neq i}(x_i-x_j)}. \]

Degree 1 polynomial, \( P(x) \), that contains \((1,3)\) and \((3,4)\)?
Work modulo 5.
\( \Delta_1(x) \) contains \((1,1)\) and \((3,0)\).
\[
\Delta_1(x) = \frac{(x-3)}{1-3} = \frac{x-3}{-2} = 2(x-3) = 2x - 6 = 2x + 4 \pmod{5}.
\]

For a quadratic, \( a_2x^2 + a_1x + a_0 \) hits \((1,3);(2,4);(3,0)\).
Work modulo 5.
Find \( \Delta_1(x) \) polynomial contains \((1,1);(2,0);(3,0)\).
\[
\Delta_1(x) = \frac{(x-2)(x-3)}{(1-2)(1-3)}
\]
Reiterating Examples.

\[ \Delta_i(x) = \frac{\prod_{j \neq i}(x-x_j)}{\prod_{j \neq i}(x_i-x_j)} \]

Degree 1 polynomial, \( P(x) \), that contains \((1, 3)\) and \((3, 4)\)?

Work modulo 5.

\( \Delta_1(x) \) contains \((1, 1)\) and \((3, 0)\).

\[ \Delta_1(x) = \frac{(x-3)}{1-3} = \frac{x-3}{-2} = 2(x - 3) = 2x - 6 = 2x + 4 \pmod{5}. \]

For a quadratic, \( a_2 x^2 + a_1 x + a_0 \) hits \((1, 3)\); \((2, 4)\); \((3, 0)\).

Work modulo 5.

Find \( \Delta_1(x) \) polynomial contains \((1, 1)\); \((2, 0)\); \((3, 0)\).

\[ \Delta_1(x) = \frac{(x-2)(x-3)}{(1-2)(1-3)} = \frac{(x-2)(x-3)}{2} \]
Reiterating Examples.

\[ \Delta_i(x) = \frac{\prod_{j \neq i}(x-x_j)}{\prod_{j \neq i}(x_i-x_j)}. \]

Degree 1 polynomial, \( P(x) \), that contains \((1,3)\) and \((3,4)\)?

Work modulo 5.

\( \Delta_1(x) \) contains \((1, 1)\) and \((3, 0)\).

\[ \Delta_1(x) = \frac{(x-3)}{1-3} = \frac{x-3}{-2} = 2(x - 3) = 2x - 6 = 2x + 4 \pmod{5}. \]

For a quadratic, \( a_2 x^2 + a_1 x + a_0 \) hits \((1, 3); (2, 4); (3, 0)\).

Work modulo 5.

Find \( \Delta_1(x) \) polynomial contains \((1, 1); (2, 0); (3, 0)\).

\[ \Delta_1(x) = \frac{(x-2)(x-3)}{(1-2)(1-3)} = \frac{(x-2)(x-3)}{2} = 3(x - 2)(x - 3) \]
Reiterating Examples.

\[ \Delta_i(x) = \frac{\prod_{j \neq i} (x-x_j)}{\prod_{j \neq i} (x_i-x_j)}. \]

Degree 1 polynomial, \( P(x) \), that contains \((1,3)\) and \((3,4)\)?

Work modulo 5.

\( \Delta_1(x) \) contains \((1,1)\) and \((3,0)\).

\[ \Delta_1(x) = \frac{(x-3)}{1-3} = \frac{x-3}{-2} = 2(x - 3) = 2x - 6 = 2x + 4 \pmod{5}. \]

For a quadratic, \( a_2 x^2 + a_1 x + a_0 \) hits \((1,3);(2,4);(3,0)\).

Work modulo 5.

Find \( \Delta_1(x) \) polynomial contains \((1,1);(2,0);(3,0)\).

\[ \Delta_1(x) = \frac{(x-2)(x-3)}{(1-2)(1-3)} = \frac{(x-2)(x-3)}{2} = 3(x - 2)(x - 3) = 3x^2 + 1 \pmod{5} \]
Reiterating Examples.

\[ \Delta_i(x) = \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x - x_j)} \cdot \]

Degree 1 polynomial, \( P(x) \), that contains \((1, 3)\) and \((3, 4)\)?

Work modulo 5.

\[ \Delta_1(x) \text{ contains } (1, 1) \text{ and } (3, 0). \]

\[ \Delta_1(x) = \frac{(x-3)}{1-3} = \frac{x-3}{-2} = 2(x - 3) = 2x - 6 = 2x + 4 \pmod{5}. \]

For a quadratic, \( a_2 x^2 + a_1 x + a_0 \) hits \((1, 3); (2, 4); (3, 0)\).

Work modulo 5.

Find \( \Delta_1(x) \) polynomial contains \((1, 1); (2, 0); (3, 0)\).

\[ \Delta_1(x) = \frac{(x-2)(x-3)}{1-2}(1-3) = \frac{(x-2)(x-3)}{2} = 3(x - 2)(x - 3) = 3x^2 + 1 \pmod{5} \]
Reiterating Examples.

\[
\Delta_i(x) = \frac{\prod_{j \neq i} (x-x_j)}{\prod_{j \neq i} (x_i-x_j)}.
\]

Degree 1 polynomial, \(P(x)\), that contains (1,3) and (3,4)?

Work modulo 5.

\(\Delta_1(x)\) contains (1,1) and (3,0).

\[
\Delta_1(x) = \frac{(x-3)}{1-3} = \frac{x-3}{-2} = 2(x - 3) = 2x - 6 = 2x + 4 \pmod{5}.
\]

For a quadratic, \(a_2 x^2 + a_1 x + a_0\) hits (1,3); (2,4); (3,0).

Work modulo 5.

Find \(\Delta_1(x)\) polynomial contains (1,1); (2,0); (3,0).

\[
\Delta_1(x) = \frac{(x-2)(x-3)}{(1-2)(1-3)} = \frac{(x-2)(x-3)}{2} = 3(x-2)(x-3)
\]

\[
= 3x^2 + 1 \pmod{5}
\]

Put the delta functions together.
For a line, $a_1 x + a_0 = mx + b$ contains points $(1,3)$ and $(2,4)$.
Simultaneous Equations Method.

For a line, \(a_1 x + a_0 = mx + b\) contains points \((1,3)\) and \((2,4)\).

\[ P(1) = \]

\[ P(2) = \]

Subtract first from second.

\[ m + b \equiv 3 \pmod{5} \]

\[ m \equiv 1 \pmod{5} \]

Backsolve:

\[ b \equiv 2 \pmod{5} \].

Secret is 2.

And the line is... \(x + 2 \pmod{5}\).
Simultaneous Equations Method.

For a line, \( a_1 x + a_0 = mx + b \) contains points \((1,3)\) and \((2,4)\).

\[
P(1) = m(1) + b \equiv m + b
\]
Simultaneous Equations Method.

For a line, $a_1 x + a_0 = mx + b$ contains points $(1,3)$ and $(2,4)$.

$$P(1) = m(1) + b \equiv m + b \equiv 3 \pmod{5}$$
Simultaneous Equations Method.

For a line, \( a_1 x + a_0 = mx + b \) contains points \((1,3)\) and \((2,4)\).

\[
P(1) = m(1) + b \equiv m + b \equiv 3 \pmod{5}
\]
\[
P(2) = m(2) + b \equiv 2m + b \equiv 4 \pmod{5}
\]
Simultaneous Equations Method.

For a line, \( a_1 x + a_0 = mx + b \) contains points \((1,3)\) and \((2,4)\).

\[
P(1) = m(1) + b \equiv m + b \equiv 3 \pmod{5}
\]
\[
P(2) = m(2) + b \equiv 2m + b \equiv 4 \pmod{5}
\]
Simultaneous Equations Method.

For a line, \( a_1 x + a_0 = mx + b \) contains points \((1,3)\) and \((2,4)\).

\[
\begin{align*}
P(1) &= m(1) + b & \equiv & & m + b & \equiv & & 3 \pmod{5} \\
P(2) &= m(2) + b & \equiv & & 2m + b & \equiv & & 4 \pmod{5}
\end{align*}
\]

Subtract first from second..
Simultaneous Equations Method.

For a line, \( a_1 x + a_0 = mx + b \) contains points \((1,3)\) and \((2,4)\).

\[
P(1) = m(1) + b \equiv m + b \equiv 3 \pmod{5}
\]
\[
P(2) = m(2) + b \equiv 2m + b \equiv 4 \pmod{5}
\]

Subtract first from second.

\[
m + b \equiv 3 \pmod{5}
\]
\[
m \equiv 1 \pmod{5}
\]

Backsolve:

\[
b \equiv 2 \pmod{5}
\]

And the line is...

\[
x + 2 \pmod{5}
\]
Simultaneous Equations Method.

For a line, $a_1 x + a_0 = mx + b$ contains points $(1,3)$ and $(2,4)$.

\[
P(1) = m(1) + b \equiv m + b \equiv 3 \pmod{5}
\]
\[
P(2) = m(2) + b \equiv 2m + b \equiv 4 \pmod{5}
\]

Subtract first from second..

\[
m + b \equiv 3 \pmod{5}
\]
\[
m \equiv 1 \pmod{5}
\]

Backsolves: $b \equiv 2 \pmod{5}$. 

Simultaneous Equations Method.

For a line, \( a_1 x + a_0 = mx + b \) contains points \((1, 3)\) and \((2, 4)\).

\[
P(1) = m(1) + b \equiv m + b \equiv 3 \pmod{5}
\]

\[
P(2) = m(2) + b \equiv 2m + b \equiv 4 \pmod{5}
\]

Subtract first from second.

\[
m + b \equiv 3 \pmod{5}
\]

\[
m \equiv 1 \pmod{5}
\]

Backsolve: \( b \equiv 2 \pmod{5} \). Secret is 2.
Simultaneous Equations Method.

For a line, $a_1 x + a_0 = mx + b$ contains points $(1,3)$ and $(2,4)$.

\[
P(1) = m(1) + b \equiv m + b \equiv 3 \pmod{5}
\]
\[
P(2) = m(2) + b \equiv 2m + b \equiv 4 \pmod{5}
\]

Subtract first from second.

\[
m + b \equiv 3 \pmod{5}
\]
\[
m \equiv 1 \pmod{5}
\]

Backsolve: $b \equiv 2 \pmod{5}$. Secret is 2.

And the line is...

\[
x + 2 \pmod{5}.
\]
Quadratic

For a quadratic polynomial, $a_2 x^2 + a_1 x + a_0$ hits $(1,2); (2,4); (3,0)$.

Plug in points to find equations.

$P(1) = a_2 + a_1 + a_0 \equiv 2 \pmod{5}$

$P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{5}$

$P(3) = 4a_2 + 3a_1 + a_0 \equiv 0 \pmod{5}$

Subtracting the second from the third yields:

$a_1 = 1$.

Thus, $a_0 = (2 - 4(1)) = -2 = 3 \pmod{5}$.

So the polynomial is $2x^2 + x + 4 \pmod{5}$.
Quadratic

For a quadratic polynomial, $a_2x^2 + a_1x + a_0$ hits $(1,2); (2,4); (3,0)$. Plug in points to find equations.
Quadratic

For a quadratic polynomial, \( a_2x^2 + a_1x + a_0 \) hits (1,2);(2,4);(3,0). Plug in points to find equations.

\[
P(1) = a_2 + a_1 + a_0 \equiv 2 \pmod{5}
\]
Quadratic

For a quadratic polynomial, $a_2 x^2 + a_1 x + a_0$ hits $(1, 2); (2, 4); (3, 0)$. Plug in points to find equations.

$$P(1) = a_2 + a_1 + a_0 \equiv 2 \pmod{5}$$
$$P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{5}$$
For a quadratic polynomial, $a_2 x^2 + a_1 x + a_0$ hits $\{1, 2\}; (2, 4); (3, 0)$. Plug in points to find equations.

\[
\begin{align*}
P(1) &= a_2 + a_1 + a_0 \equiv 2 \pmod{5} \\
P(2) &= 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{5} \\
P(3) &= 4a_2 + 3a_1 + a_0 \equiv 0 \pmod{5}
\end{align*}
\]
For a quadratic polynomial, \( a_2 x^2 + a_1 x + a_0 \) hits \((1,2); (2,4); (3,0)\). Plug in points to find equations.

\[
P(1) = a_2 + a_1 + a_0 \equiv 2 \pmod{5} \\
P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{5} \\
P(3) = 4a_2 + 3a_1 + a_0 \equiv 0 \pmod{5}
\]
For a quadratic polynomial, \( a_2 x^2 + a_1 x + a_0 \) hits \((1, 2); (2, 4); (3, 0)\). Plug in points to find equations.

\[
P(1) = a_2 + a_1 + a_0 \equiv 2 \, (\text{mod} \, 5) \\
P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \, (\text{mod} \, 5) \\
P(3) = 4a_2 + 3a_1 + a_0 \equiv 0 \, (\text{mod} \, 5)
\]

\[
a_2 + a_1 + a_0 \equiv 2 \, (\text{mod} \, 5) \\
3a_1 + 2a_0 \equiv 1 \, (\text{mod} \, 5) \\
4a_1 + 2a_0 \equiv 2 \, (\text{mod} \, 5)
\]
For a quadratic polynomial, $a_2 x^2 + a_1 x + a_0$ hits $(1, 2); (2, 4); (3, 0)$. Plug in points to find equations.

\[
P(1) = a_2 + a_1 + a_0 \equiv 2 \pmod{5}
\]

\[
P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{5}
\]

\[
P(3) = 4a_2 + 3a_1 + a_0 \equiv 0 \pmod{5}
\]

\[
a_2 + a_1 + a_0 \equiv 2 \pmod{5}
\]

\[
3a_1 + 2a_0 \equiv 1 \pmod{5}
\]

\[
4a_1 + 2a_0 \equiv 2 \pmod{5}
\]

Subtracting 2nd from 3rd yields: $a_1 = 1$. 
For a quadratic polynomial, $a_2x^2 + a_1x + a_0$ hits (1,2);(2,4);(3,0). Plug in points to find equations.

\[
P(1) = a_2 + a_1 + a_0 \equiv 2 \pmod{5}
\]
\[
P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{5}
\]
\[
P(3) = 4a_2 + 3a_1 + a_0 \equiv 0 \pmod{5}
\]

\[
a_2 + a_1 + a_0 \equiv 2 \pmod{5}
\]
\[
3a_1 + 2a_0 \equiv 1 \pmod{5}
\]
\[
4a_1 + 2a_0 \equiv 2 \pmod{5}
\]

Subtracting 2nd from 3rd yields: $a_1 = 1$.
$a_0 = (2 - 4(a_1))2^{-1}$
For a quadratic polynomial, $a_2x^2 + a_1x + a_0$ hits $(1,2); (2,4); (3,0)$. Plug in points to find equations.

\[
\begin{align*}
P(1) &= a_2 + a_1 + a_0 \equiv 2 \pmod{5} \\
P(2) &= 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{5} \\
P(3) &= 4a_2 + 3a_1 + a_0 \equiv 0 \pmod{5}
\end{align*}
\]

\[
\begin{align*}
a_2 + a_1 + a_0 &\equiv 2 \pmod{5} \\
3a_1 + 2a_0 &\equiv 1 \pmod{5} \\
4a_1 + 2a_0 &\equiv 2 \pmod{5}
\end{align*}
\]

Subtracting 2nd from 3rd yields: $a_1 = 1$.

\[
a_0 = (2 - 4(a_1))2^{-1} = (-2)(2^{-1})
\]
For a quadratic polynomial, $a_2x^2 + a_1x + a_0$ hits $(1, 2); (2, 4); (3, 0)$. Plug in points to find equations.

\[
P(1) = a_2 + a_1 + a_0 \equiv 2 \pmod{5}
\]
\[
P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{5}
\]
\[
P(3) = 4a_2 + 3a_1 + a_0 \equiv 0 \pmod{5}
\]

\[
a_2 + a_1 + a_0 \equiv 2 \pmod{5}
\]
\[
3a_1 + 2a_0 \equiv 1 \pmod{5}
\]
\[
4a_1 + 2a_0 \equiv 2 \pmod{5}
\]

Subtracting 2nd from 3rd yields: $a_1 = 1$.
$a_0 = (2 - 4(a_1))2^{-1} = (-2)(2^{-1}) = (3)(3)$
Quadratic

For a quadratic polynomial, \( a_2 x^2 + a_1 x + a_0 \) hits \((1, 2); (2, 4); (3, 0)\). Plug in points to find equations.

\[
P(1) = a_2 + a_1 + a_0 \equiv 2 \pmod{5}
\]
\[
P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{5}
\]
\[
P(3) = 4a_2 + 3a_1 + a_0 \equiv 0 \pmod{5}
\]

\[
a_2 + a_1 + a_0 \equiv 2 \pmod{5}
\]
\[
3a_1 + 2a_0 \equiv 1 \pmod{5}
\]
\[
4a_1 + 2a_0 \equiv 2 \pmod{5}
\]

Subtracting 2nd from 3rd yields: \( a_1 = 1 \).

\[
a_0 = (2 - 4(a_1))2^{-1} = (-2)(2^{-1}) = (3)(3) = 9 \equiv 4 \pmod{5}
\]
For a quadratic polynomial, \( a_2x^2 + a_1x + a_0 \) hits \((1,2);(2,4);(3,0)\). Plug in points to find equations.

\[
P(1) = a_2 + a_1 + a_0 \equiv 2 \pmod{5}
\]
\[
P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{5}
\]
\[
P(3) = 4a_2 + 3a_1 + a_0 \equiv 0 \pmod{5}
\]

\[
a_2 + a_1 + a_0 \equiv 2 \pmod{5}
\]
\[
3a_1 + 2a_0 \equiv 1 \pmod{5}
\]
\[
4a_1 + 2a_0 \equiv 2 \pmod{5}
\]

Subtracting 2nd from 3rd yields: \( a_1 = 1 \).

\[
a_0 = (2 - 4(a_1))2^{-1} = (-2)(2^{-1}) = (3)(3) = 9 \equiv 4 \pmod{5}
\]
\[
a_2 = 2 - 1 - 4 \equiv 2 \pmod{5}
\]
For a quadratic polynomial, \( a_2 x^2 + a_1 x + a_0 \) hits \((1,2); (2,4); (3,0)\). Plug in points to find equations.

\[
P(1) = a_2 + a_1 + a_0 \equiv 2 \pmod{5}
\]
\[
P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{5}
\]
\[
P(3) = 4a_2 + 3a_1 + a_0 \equiv 0 \pmod{5}
\]

\[
a_2 + a_1 + a_0 \equiv 2 \pmod{5}
\]
\[
3a_1 + 2a_0 \equiv 1 \pmod{5}
\]
\[
4a_1 + 2a_0 \equiv 2 \pmod{5}
\]

Subtracting 2nd from 3rd yields: \( a_1 = 1 \).
\[
a_0 = (2 - 4(a_1))2^{-1} = (-2)(2^{-1}) = (3)(3) = 9 \equiv 4 \pmod{5}
\]
\[
a_2 = 2 - 1 - 4 \equiv 2 \pmod{5}
\]
For a quadratic polynomial, $a_2x^2 + a_1x + a_0$ hits $(1, 2); (2, 4); (3, 0)$. Plug in points to find equations.

\[
P(1) = a_2 + a_1 + a_0 \equiv 2 \pmod{5}
\]
\[
P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{5}
\]
\[
P(3) = 4a_2 + 3a_1 + a_0 \equiv 0 \pmod{5}
\]

Subtracting 2nd from 3rd yields: $a_1 = 1$. 

\[
a_0 = (2 - 4(a_1))2^{-1} = (-2)(2^{-1}) = (3)(3) = 9 \equiv 4 \pmod{5}
\]
\[
a_2 = 2 - 1 - 4 \equiv 2 \pmod{5}.
\]

So polynomial is $2x^2 + 1x + 4 \pmod{5}$
In general...

Given points: \((x_1, y_1); (x_2, y_2) \ldots (x_k, y_k)\).
In general..

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

Solve...

\[
a_{k-1}x_1^{k-1} + \cdots + a_0 \equiv y_1 \pmod{p}
\]
\[
a_{k-1}x_2^{k-1} + \cdots + a_0 \equiv y_2 \pmod{p}
\]
\[
\vdots \quad \vdots \quad \vdots
\]
\[
a_{k-1}x_k^{k-1} + \cdots + a_0 \equiv y_k \pmod{p}
\]

Will this always work?

As long as solution exists and it is unique!

Modular Arithmetic Fact:

Exactly 1 polynomial of degree \(\leq d\) with arithmetic modulo prime contains \(d+1\) pts.
In general..

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

Solve...

\[
a_{k-1}x_1^{k-1} + \cdots + a_0 \equiv y_1 \pmod{p}
\]
\[
a_{k-1}x_2^{k-1} + \cdots + a_0 \equiv y_2 \pmod{p}
\]
\[
\vdots \quad \vdots \quad \vdots
\]
\[
a_{k-1}x_k^{k-1} + \cdots + a_0 \equiv y_k \pmod{p}
\]

Will this always work?
In general..

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

Solve...

\[
\begin{align*}
  a_{k-1}x_1^{k-1} + \cdots + a_0 & \equiv y_1 \pmod{p} \\
  a_{k-1}x_2^{k-1} + \cdots + a_0 & \equiv y_2 \pmod{p} \\
  \vdots & \vdots \vdots \\
  a_{k-1}x_k^{k-1} + \cdots + a_0 & \equiv y_k \pmod{p}
\end{align*}
\]

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...

\[
a_{k-1}x_1^{k-1} + \cdots + a_0 \equiv y_1 \pmod{p}
\]

\[
a_{k-1}x_2^{k-1} + \cdots + a_0 \equiv y_2 \pmod{p}
\]

\[
\vdots \quad \vdots \quad \vdots
\]

\[
a_{k-1}x_k^{k-1} + \cdots + a_0 \equiv y_k \pmod{p}
\]

Will this always work?

As long as solution \textbf{exists} and it is \textbf{unique}! And...
In general..

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

Solve...

\[
a_{k-1}x_1^k + \cdots + a_0 \equiv y_1 \pmod{p}
\]
\[
a_{k-1}x_2^k + \cdots + a_0 \equiv y_2 \pmod{p}
\]
\[
\vdots \hspace{1cm} \vdots \hspace{1cm} \vdots
\]
\[
a_{k-1}x_k^k + \cdots + a_0 \equiv y_k \pmod{p}
\]

Will this always work?

As long as solution \textbf{exists} and it is \textbf{unique}! And...

\textbf{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.
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: (proved last time)
**Summary.**

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

Existence:
- Lagrange Interpolation.

Uniqueness: (proved last time)
- 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.
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 Revisited

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

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

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

Efficiency: ???
Secret Sharing Revisited

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

**Robustness:** Any \( k \) knows secret.
Secret Sharing Revisited

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

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

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

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

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

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

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

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

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

**Efficiency:** ???
Secret Sharing Revisited

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

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

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

**Efficiency:** ???
Efficiency.

For a $b$-bit secret, must choose a prime $p > 2^b$. Theorem: There is always a prime between $n$ and $2^n$. 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.
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 $p > 2^b$. 
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 $2n$. 

Efficiency.
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 $2n$.

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 $2n$.
Working over numbers within 1 bit of secret size.
**Minimal!**
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 $2n$. 
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: $P(1) \ldots P(n)$.
For $b$-bit secret, must choose a prime $p > 2^b$.

**Theorem:** 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 $2n$.

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

**Theorem:** 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: $P(1) \ldots P(n)$.
For $b$-bit secret, must choose a prime $p > 2^b$.

**Theorem:** 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.

1. Evaluate degree $n - 1$ polynomial $n + k$ times using $\log_2 p$-bit numbers. $O\left(\frac{kn \log_2 p}{\log_2 n}\right)$.

2. Reconstruct secret by solving system of $n$ equations using $\log_2 p$-bit arithmetic. $O\left(n^3 \log_2 p\right)$.

3. Matrix has special form so $O\left(n \log n \log_2 p\right)$ reconstruction. Faster versions in practice are almost as efficient.
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: 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. $O(n^3 \log^2 p)$.

Faster versions in practice are almost as efficient.
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.
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.
A bit of counting.

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

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

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

Infinite number for reals, rationals, complex numbers!
A bit of 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 of 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!
Erasure Codes.

Satellite

GPS device
Erasure Codes.

Satellite

3 packet message.

GPS device
Erasure Codes.

Satellite

3 packet message.

GPS device

Lose 3 out 6 packets.
Erasure Codes.

Satellite

1 2 3 1 2 3

GPS device

3 packet message. So send 6!

Lose 3 out 6 packets.
Erasure Codes.

Satellite

3 packet message. So send 6!

GPS device

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

**Channel:** Lossy channel: loses \( k \) packets.
Problem: Want to send a message with $n$ packets.
Channel: Lossy channel: loses $k$ packets.
Question: Can you send $n + k$ packets and recover message?
Problem: Want to send a message with $n$ packets.

Channel: Lossy channel: loses $k$ packets.

Question: Can you send $n + k$ packets and recover message?

Solution Idea: Use Polynomials!!!
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 <i>k</i> packets.

Must send <i>n + k</i> packets!

Any <b>n packets</b> should allow reconstruction of <i>n packet message</i>.

Any <b>n point values</b> allow reconstruction of degree <i>n − 1</i> 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 which has $n$ coefficients!
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 which has $n$ coefficients!

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 which has $n$ coefficients!

Alright!!

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

Channel: Lossy channel: loses $k$ packets.

Question: Can you send $n + k$ packets and recover message?
**Problem:** Want to send a message with \( n \) packets.

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

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

A degree \( n - 1 \) polynomial determined by any \( n \) points!
**Problem:** Want to send a message with $n$ packets.

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

**Question:** Can you send $n+k$ packets and recover message?

A degree $n−1$ polynomial determined by any $n$ points!

Erasure Coding Scheme: message = $m_0, m_1, m_2, \ldots, m_{n−1}$. Each $m_i$ is a packet.
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 = \( m_0, m_1, m_2, \ldots, m_{n-1} \). Each \( m_j \)
is a packet.

1. Choose prime \( p > 2^b \) for packet size \( b \) (size = number of bits).
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 = \( m_0, m_1, m_2, \ldots, m_{n-1} \). Each \( m_i \) is a packet.

1. Choose prime \( p > 2^b \) for packet size \( b \) (size = number of bits).
2. \( P(x) = m_{n-1}x^{n-1} + \cdots m_0 \pmod p \).
**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 = $m_0, m_1, m_2, \ldots, m_{n-1}$. Each $m_i$ is a packet.

1. Choose prime $p > 2^b$ for packet size $b$ (size = number of bits).
2. $P(x) = m_{n-1}x^{n-1} + \cdots m_0 \pmod{p}$.
3. Send $P(1), \ldots, P(n+k)$. 
**Problem:** Want to send a message with \( n \) packets.

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

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

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

Erasure Coding Scheme: message = \( m_0, m_1, m_2, \ldots, m_{n-1} \). Each \( m_j \) is a packet.

1. Choose prime \( p > 2^b \) for packet size \( b \) (size = number of bits).
2. \( P(x) = m_{n-1}x^{n-1} + \cdots + m_0 \mod p \).
3. Send \( P(1), \ldots, P(n+k) \).
**Problem:** Want to send a message with \( n \) packets.

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

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

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

Erasure Coding Scheme: message = \( m_0, m_1, m_2, \ldots, m_{n-1} \). Each \( m_j \) is a packet.

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

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

Channel: Lossy channel: loses $k$ packets.

Question: Can you send $n + k$ packets and recover message?

A degree $n - 1$ polynomial determined by any $n$ points!

Erasure Coding Scheme: message $= m_0, m_1, m_2, \ldots, m_{n-1}$. Each $m_i$ is a packet.

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

Any $n$ of the $n + k$ packets gives polynomial ...and message!
Erasure Codes.

Satellite

GPS device
Erasure Codes.

Satellite

GPS device

$n$ packet message.
Erasure Codes.

Satellite

$n$ packet message.

Lose $k$ packets.

GPS device

Any $n$ packets is enough!
Erasure Codes.

Satellite

1 2 ······ n+k

GPS device

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

Lose $k$ packets.
Erasure Codes.

Satellite

1 2 \cdots n+k

GPS device

\textit{n} packet message. So send \( n+k \)!

Lose \( k \) packets.
Erasure Codes.

1 2 … \(n+k\)

Satellite

\[\begin{array}{c}
1 \\
2 \\
\vdots \\
\hline
\end{array}\]

GPS device

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

Lose \(k\) packets.
Erasure Codes.

Satellite

1 2 \ldots n+k

GPS device

\begin{itemize}
\item Lose $k$ packets.
\item Any $n$ packets is enough!
\end{itemize}

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

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

Lose $k$ packets.

Any $n$ packets is enough!
Erasure Codes.

Satellite

1 2 ... n+k

n packet message. So send n+k!

Lose k packets.

Any n packets is enough!

n packet message.

Optimal.
Comparison with Secret Sharing.

Comparing information content:
Comparison with Secret Sharing.

Comparing information content:

Secret Sharing: each share is size of whole secret.
Comparison with Secret Sharing.

Comparing 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. up to 3 erasures.

\[ n = 3, \quad k = 3 \]

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

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

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

Better work modulo 7 at least!

Why? \((0, P(0)) = (5, P(5)) \mod 5\).
Erasure Code: Example.

Send message of 1,4, and 4. up to 3 erasures. $n = 3, k = 3$

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 $(0, P(0)) ... (5, P(5))$. 6 points. Better work modulo 7 at least!

Why? $(0, P(0)) = (5, P(5)) \pmod{5}$
Erasure Code: Example.

Send message of 1, 4, and 4. up to 3 erasures. \( n = 3, k = 3 \)

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

Send message of 1, 4, and 4. up to 3 erasures. \( n = 3, k = 3 \)

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

How?
Erasure Code: Example.

Send message of 1, 4, and 4. up to 3 erasures. $n = 3, k = 3$

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

How?

Lagrange Interpolation.
Send message of 1,4, and 4. up to 3 erasures. \( n = 3,k = 3 \)

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

How?

Lagrange Interpolation.
Linear System.
Send message of 1, 4, and 4. up to 3 erasures. $n = 3, k = 3$

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. up to 3 erasures. \( n = 3, k = 3 \)

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. up to 3 erasures. \( n = 3, k = 3 \)

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

How?

Lagrange Interpolation.
Linear System.

Work modulo 5.

\[ P(x) = x^2 \pmod{5} \]
Send message of 1, 4, and 4. up to 3 erasures. $n = 3, k = 3$

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. up to 3 erasures. \( n = 3, k = 3 \)

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

How?

Lagrange Interpolation.
Linear System.

Work modulo 5.

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

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

Better work modulo 7 at least!

Why?

\[
(P(0), P(0)) = (5, P(5)) \pmod{5}
\]
Send message of 1, 4, and 4. up to 3 erasures. $n = 3, k = 3$

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 message of 1, 4, and 4. up to 3 erasures. \( n = 3, k = 3 \)

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. up to 3 erasures. \( n = 3, k = 3 \)

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

How?

Lagrange Interpolation.
Linear System.

Work modulo 5.

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

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

Send \((0, P(0)) \ldots (5, P(5))\).
Send message of 1, 4, and 4. up to 3 erasures. \( n = 3, k = 3 \)

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. up to 3 erasures. \( n = 3, k = 3 \)

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. up to 3 erasures. \( n = 3, k = 3 \)

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. up to 3 erasures. \( n = 3, k = 3 \)

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. up to 3 erasures. \( n = 3, k = 3 \)

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:

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

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

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

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

Notice that packets contain “x-values”.
Example

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

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

Send Packets: $(1, 1), (2, 4), (3, 4), (4, 7), (5, 2), (6, 0)$
Notice that packets contain “x-values”.
**Example**

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

Modulo 7 to accommodate at least 6 packets.

Linear equations:

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

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

Make polynomial with \( P(1) = 1, \ P(2) = 4, \ P(3) = 4. \)
Modulo 7 to accommodate at least 6 packets.
Linear equations:
\[
\begin{align*}
P(1) &= a_2 + a_1 + a_0 \equiv 1 \pmod{7} \\
P(2) &= 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \\
P(3) &= 2a_2 + 3a_1 + a_0 \equiv 4 \pmod{7} \\
\end{align*}
\]
\[6a_1 + 3a_0 = 2 \pmod{7}, \ 5a_1 + 4a_0 = 0 \pmod{7}\]
Example

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Linear equations:

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

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

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

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

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

Send
Example

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

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

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

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

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

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

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

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

Linear equations:

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

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

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

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

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

Send

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

Notice that packets contain “x-values”. 
Summary: Polynomials are useful!
Summary: Polynomials are useful!

- give Secret Sharing.
Summary: Polynomials are useful!

- give Secret Sharing.
- give Erasure Codes.
Summary: Polynomials are useful!

- give Secret Sharing.
- give Erasure Codes.

Next time: correct broader class of errors!