Secret Sharing

Today.

Last time:
Shared (and sort of kept) secrets.
Today: Errors
Tolerate Loss: erasure codes.
Tolerate corruption!

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.
In the rationals, the precision blows up, where in modular arithmetic, it does not.

Secret Sharing

Properties

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

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

Uniqueness:

Property 1 A non-zero degree \( d \) polynomial has at most \( d \) roots.
Factoring: \( P(x) \) with roots \( r_1, \ldots, r_d \)
\[ P(x) = c(x - r_1)(x - r_2) \cdots (x - r_d). \]
Love me some contradiction!
Two polynomials: \( P(x), Q(x), P(x) - Q(x) \) has too many roots.

Proof sketches.

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

In general..

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

\[
\begin{align*}
ax_1^{k-1} + \cdots + a_0 &= y_1 \pmod{p} \\
ax_2^{k-1} + \cdots + a_0 &= y_2 \pmod{p} \\
&\vdots \\
ax_k^{k-1} + \cdots + a_0 &= y_k \pmod{p}
\end{align*}
\]

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.

Secret Sharing

Modular Arithmetic Fact: Exactly one polynomial degree \( \leq d \) over \( GF(p) \), 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 \( P(x) = a_0 x^{k-1} + a_1 x^{k-2} + \cdots + a_{k-1} \) with \( a_0 = s \).
3. Share \( i \) is point \((i, P(i) \pmod{p})\).

Robustness: Any \( k \) knows secret.
Knowing \( k \) pts, only one \( P(x) \), evaluate \( P(0) \).
Secrecy: Any \( k - 1 \) knows nothing.
Knowing \( \leq k - 1 \) pts, any \( P(0) \) is possible.

Two points make a line: the value of one point allows any y-intercept.
3 kids hand out 3 points. Any two know the line.
Erasure Codes.

Satellite
GPS device

3 packet message. So send 6!

Satellite
GPS device

3 packet message. So send 6!

Lose 3 out 6 packets.

1 2 3 1 2 3
G gets packets 1,1, and 3.

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.

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

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

Any $n$ of the $n + k$ packets gives polynomial ...and 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 $GF(2^b)$ 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.

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

Make polynomial with \( P(1) = 1, P(2) = 4, P(3) = 4 \).
Modulo 7 to accommodate at least 6 packets.
Linear equations:
\[
\begin{align*}
P(1) = a_2 + a_1 + a_0 &= 1 \pmod{7} \\
P(2) = 4a_2 + 2a_1 + a_0 &= 4 \pmod{7} \\
P(3) = 2a_2 + 3a_1 + a_0 &= 4 \pmod{7}
\end{align*}
\]
6a_1 + 3a_0 = 2 \pmod{7}, 5a_0 + 4a_0 = 0 \pmod{7}
\( a_1 = 2a_0, \ 6a_1 + a_0 = 2 \pmod{7}, a_1 = 4 \pmod{7}, a_2 = 2 \pmod{7} \)
\( P(x) = 2x^2 + 4x + 2 \)
\( P(1) = 1, P(2) = 4, \) 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)
Recieve: (1,1), (2,4), (6,0)
Reconstruct?
Format: (i, R(i)).
Lagrange or linear equations.
\[
\begin{align*}
P(1) = a_2 + a_1 + a_0 &= 1 \pmod{7} \\
P(2) = 4a_2 + 2a_1 + a_0 &= 4 \pmod{7} \\
P(6) = 2a_2 + 3a_1 + a_0 &= 0 \pmod{7}
\end{align*}
\]
Channeling Sahai ...
\( P(x) = 2x^2 + 4x + 2 \)
Message? \( P(1) = 1, P(2) = 4, P(3) = 4 \).

Questions for Review

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

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

Error Correction:

Satellite

3 packet message. Send 5.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Corrupts 1 packets.

<table>
<thead>
<tr>
<th>A</th>
<th>B'</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

GPS device

The Scheme.

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

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

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

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

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

\[ Q(x) = R(i), \text{ on set of size } n+k. \]
\[ P(x) = R(i), \text{ on set of size } n+k. \]

Only \( n+k \) points total. Sets can differ by at most \( k \).
\[ \Rightarrow P(i) = R(i) = Q(i) \text{ on } \geq n \text{ values of } i. \]
\[ \Rightarrow Q(i) = P(i) \text{ at } n \text{ points.} \]
\[ \Rightarrow Q(x) = P(x). \]

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 \( n+k = 3 + 1 = 4 \) points.

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

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

In general...

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

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

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

How do we find where the bad packets are efficiently?!!??!
Where oh where can my bad packets be?

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

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

- \( E(x) \) has degree \( k \)

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

\( \Rightarrow \) \( k \) (unknown) coefficients. Leading coefficient is 1.

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

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

\( \Rightarrow \) \( n+k \) (unknown) coefficients.

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

Example.

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

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

Plug points:

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

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

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

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

4 unknowns \( (p_0, p_1, p_2, p_0) \) and \( E(x) \), \( 5 \) nonlinear equations.

Example.

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

..turn their heads each day,

Equations:

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

and linear in \( a_i \) and coefficients of \( E(x) \)!

Example.

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

\[ Q(x) = E(x)P(x) = a_{n+k-1}x^{n+k-1} + \cdots + a_0 \]
\[ E(x) = x + b_0 \]
\[ Q(i) = R(i)E(i) \]

Find \( Q(x) = Q(x)/E(x) \).
Example: finishing up.

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

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

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

What is \( x-2 \)? Hole there?

Error Correction: Berlekamp-Welsh

Message: \( m_1, \ldots, m_n. \)

Sender:
1. Form degree \( n - 1 \) polynomial \( P(x) \) where \( P(i) = m_i. \)
2. Send \( P(1), \ldots, P(n+2k). \)

Receiver:
1. Receive \( R(1), \ldots, R(n+2k). \)
2. Solve \( n+2k \) equations, \( Q(i) = E(i)R(i) \) to find \( Q(x) = E(x)P(x) \) and \( E(x). \)
3. Compute \( P(x) = Q(x)/E(x). \)
4. Compute \( P(1), \ldots, P(n). \)

Check your understanding.

You have error locator polynomial!
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.

Unique solution for \( P(x) \)

Uniqueness: any solution \( Q'(x) \) and \( E'(x) \) have
\[
\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)}. \tag{1}
\]

Proof:
We claim
\[
\frac{Q'(x)}{E'(x)} = Q(x)/E(x) \text{ on } n + 2k \text{ values of } x. \tag{2}
\]
Equation 2 implies 1:
\[ Q'(x)E(x) \text{ and } Q(x)E'(x) \text{ are degree } n + 2k - 1 \]
and agree on \( n + 2k \) points
\[ E(x) \text{ and } E'(x) \text{ have at most } k \text{ zeros each.} \]
Can cross divide at \( n \) points.
\[ \frac{Q'(x)}{E'(x)} - \frac{Q(x)}{E(x)} \text{ equal on } n \text{ points.} \]
Both degree \( \leq n - 1 \) \implies Same polynomial!

Fact: \( Q'(x)E(x) = Q(x)E'(x) \) on \( n + 2k \) values of \( x. \)
Proof: Construction implies that
\[ Q(i) = R(i)E(i) \]
\[ Q'(i) = R(i)E'(i) \]
for \( i \in \{1, \ldots, n+2k\}. \)
If \( E(i) = 0, \) then \( Q(i) = 0. \) If \( E'(i) = 0, \) then \( Q'(i) = 0. \)
\[ \implies Q(i)/E(i) = Q'(i)/E'(i) \text{ holds when } E(i) \text{ or } E'(i) \text{ are zero.} \]
When \( E'(i) \) and \( E(i) \) are not zero
\[ \frac{Q'(i)}{E'(i)} - \frac{Q(i)}{E(i)} = R(i). \]
Cross multiplying gives equality in fact for these points.

Last bit.

Points to polynomials, have to deal with zeros!
Example: dealing with \( \frac{x}{x-2} \) at \( x = 2. \)
Poll

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

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

Packets 1 and 4 are corrupted.

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

(E) is false.

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

(A), (C), (E), (F) doesn’t type check!

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!