Fast n Furious Transforms

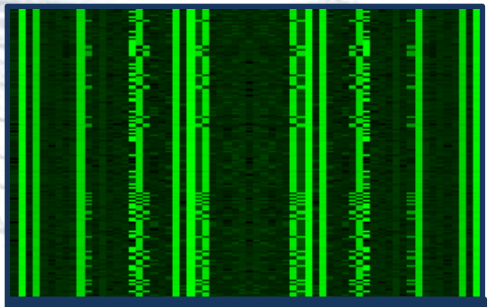
Welcome to my Journey

Pitch Detection

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, \ldots, N-1$$

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} \qquad n = 0, \ldots, N-1.$$

Fourier Transform



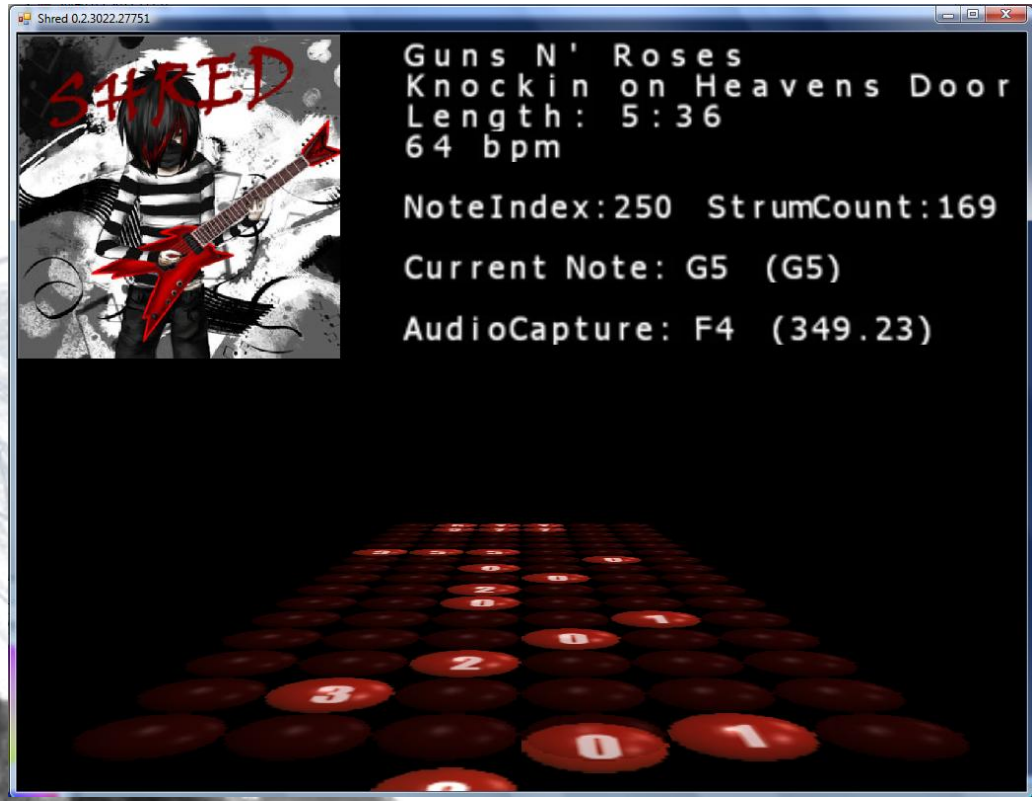
Signal Visualization

```
struct _unknown
{
    byte typeField;
    short lenField;
    char[] dataField;
}
```

Protocol Analysis

SHRED

- SHRED is a proof of concept music education software tool



SHRED

- SHRED is the first software of its kind that it uses real-time signal processing to analyze a live performance

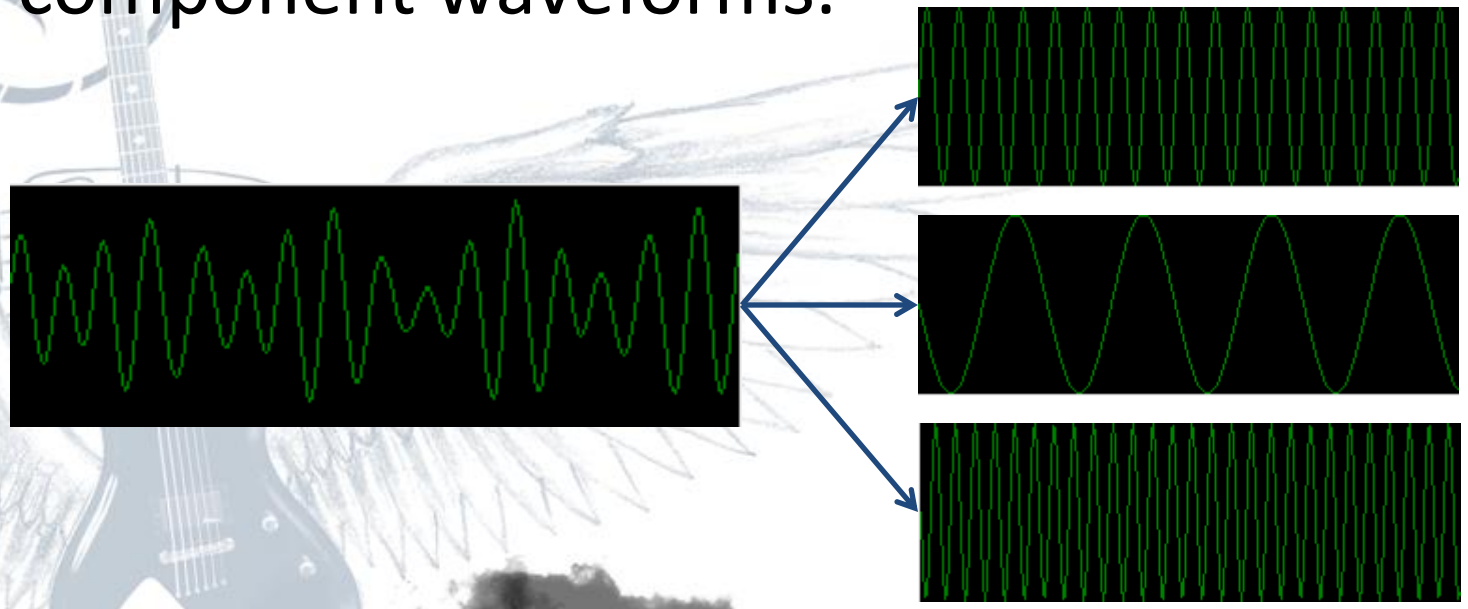
- Use real musical instruments or voice as an input device

How SHRED Works

- SHRED parses standard tablature file formats to obtain fret board finger positions
- Chords are synthesized internally to determine the chord's pitch
- Microphone audio is processed real-time to determine the player's accuracy

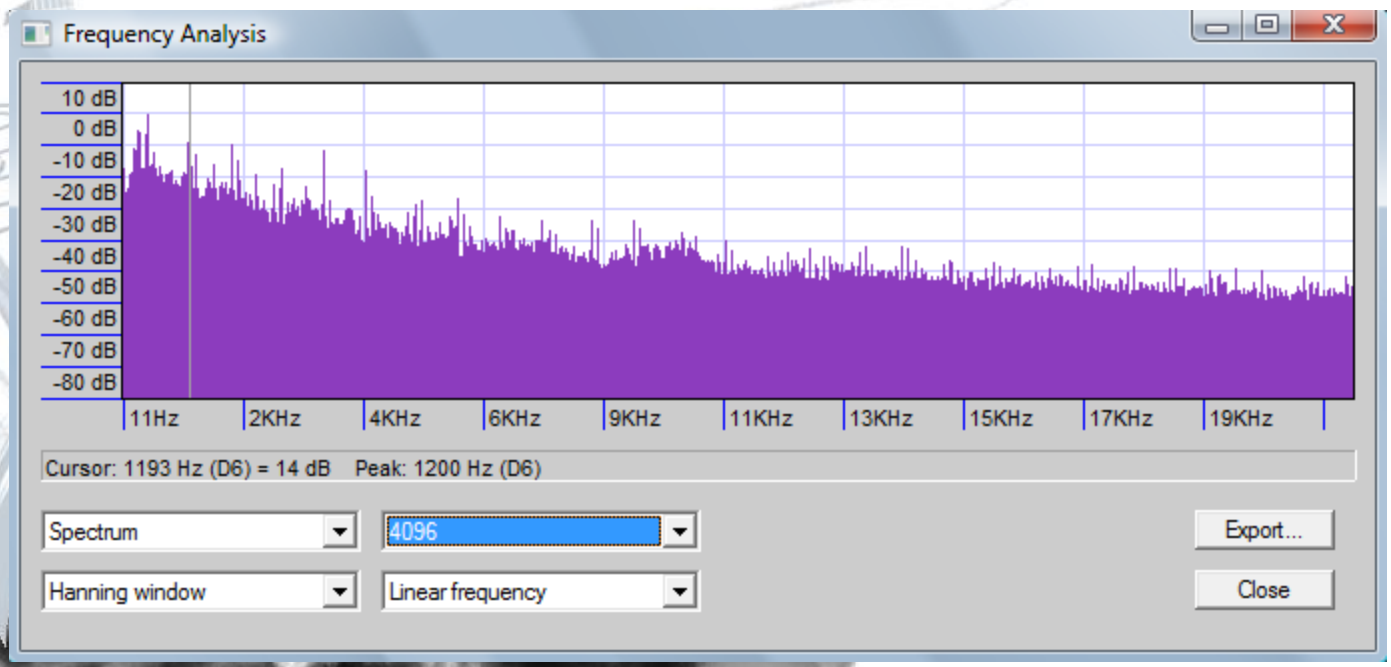
Fourier Analysis

- Fourier Analysis is a technique for decomposing compound signals into their component waveforms.



Fourier Transform

- Samples of waves captured over a period of time can be converted into their frequency spectrum via a Fourier Transform.



Discrete Fourier Transform

- The Fourier Transform attempts to detect the presence of a frequency by multiplying the original wave with the basis wave function

```
int N = input.Length;
ComplexF[] output = new ComplexF[N];
double sinWave, cosWave;

for (int k = 0; k < N; k++)
{
   sinWave = cosWave = 0;
   for (int n = 0; n < N; n++)
   {
      cosWave += input[n].Re * Math.Cos(-2 * Math.PI * k * n / N);
      sinWave += input[n].Re * Math.Sin(-2 * Math.PI * k * n / N);
   }
   // Magnitude
   output[k] = (ComplexF)Math.Sqrt(Math.Pow(cosWave, 2) + Math.Pow(sinWave, 2));
}
```

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, \ldots, N-1$$

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} \qquad n = 0, \ldots, N-1.$$

Discrete Fourier Transform

- The sum of the products from the multiplied wave represents the magnitude or overall presence of the basis wave.

```
int N = input.Length;
ComplexF[] output = new ComplexF[N];
double sinWave, cosWave;

for (int k = 0; k < N; k++)
{
   sinWave = cosWave = 0;
   for (int n = 0; n < N; n++)
   {
      cosWave += input[n].Re * Math.Cos(-2 * Math.PI * k * n / N);
      sinWave += input[n].Re * Math.Sin(-2 * Math.PI * k * n / N);
   }
   // Magnitude
   output[k] = (ComplexF)Math.Sqrt(Math.Pow(cosWave, 2) + Math.Pow(sinWave, 2));
}
```

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, \ldots, N-1$$

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} \qquad n = 0, \ldots, N-1.$$

Fast Fourier Transform

- Any optimized implementation of the DFT

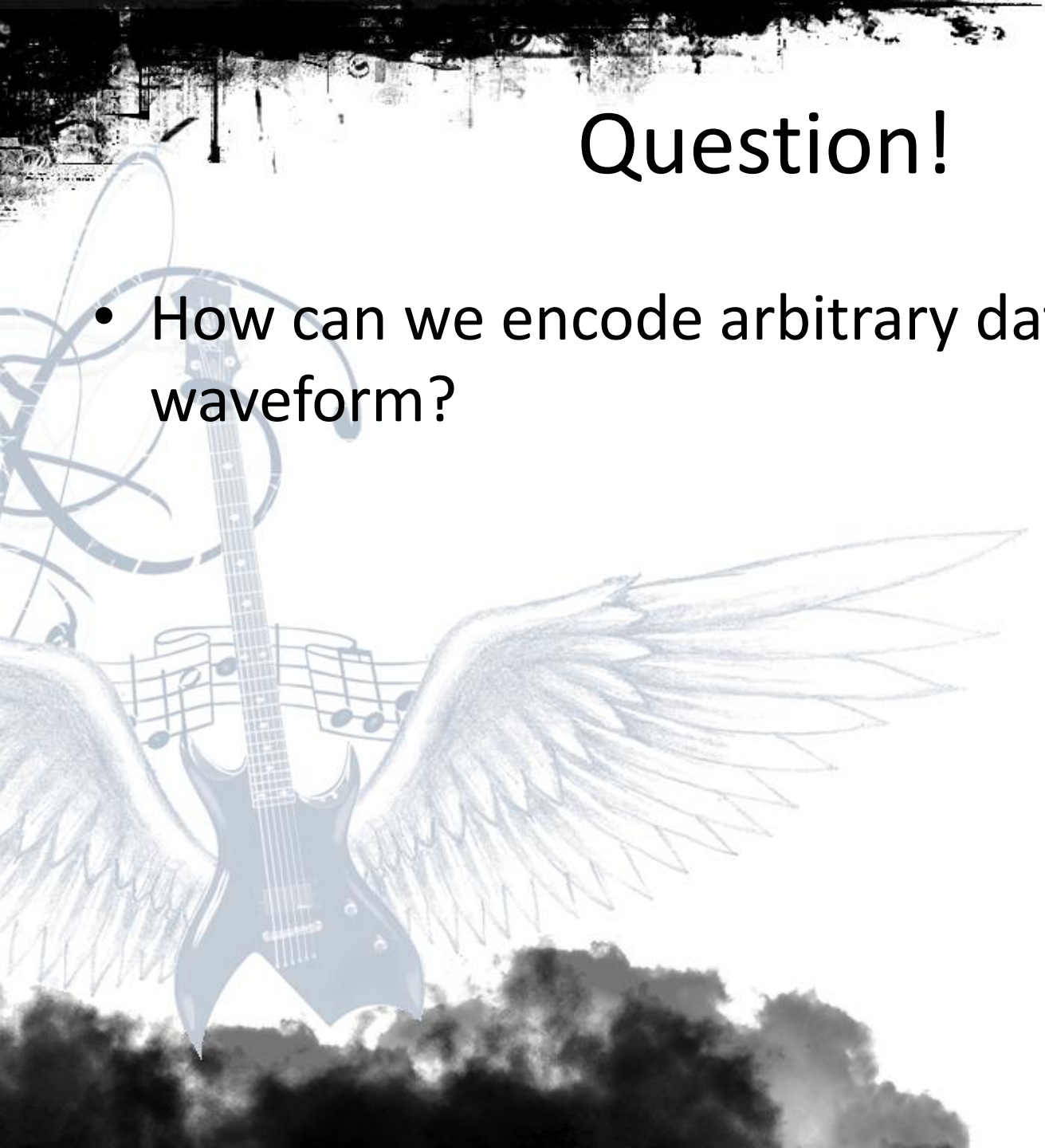
DFT complexity: $O(N)^2$

FFT complexity: $O(N \log N)$

- Example: Cooley-Turkey
  - Recursively breaks down DFT into smaller DFTs
  - Number of samples must be factorable

Question!

- How can we encode arbitrary data as a waveform?



Noise in the Data

- Sine wave:

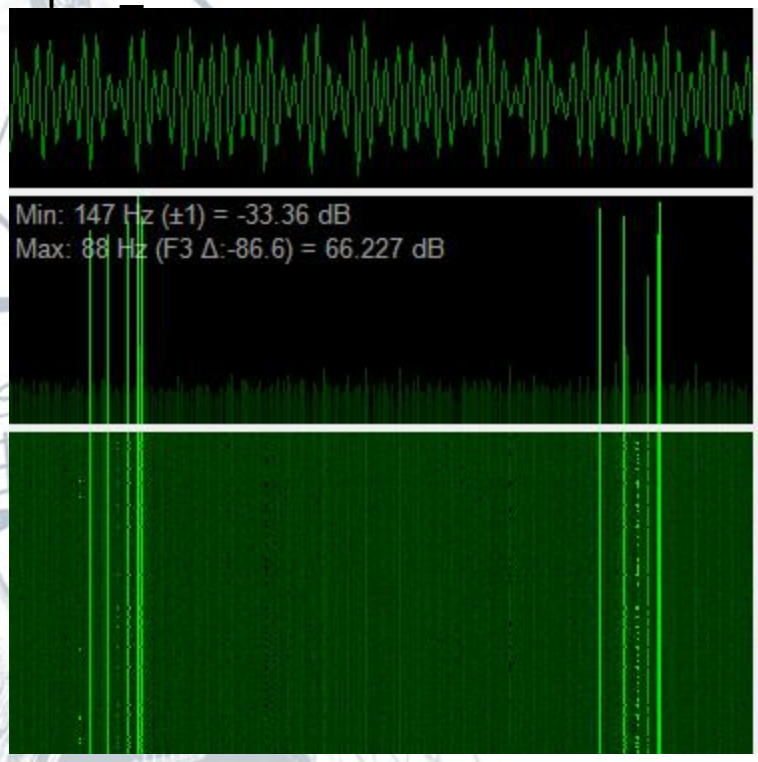
$$y(t) = A \cdot \sin(\omega t + \theta)$$

```
double theta = (2.0 * Math.PI) * (_frequency / _samplingRate);
values[i] = _amplitude * Math.Sin(theta * i + phase);
```

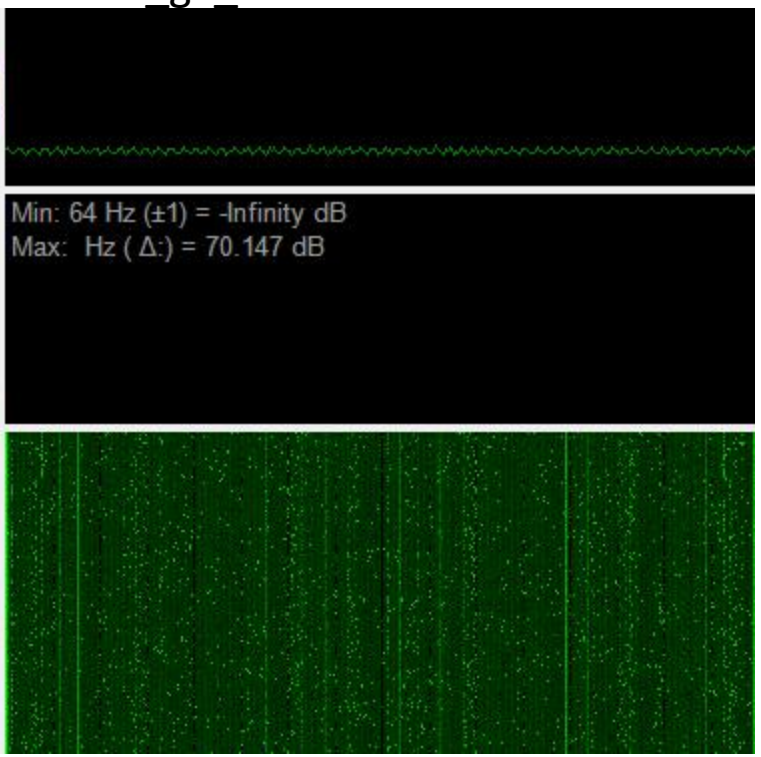
- Direct encoding:
  - Each basis wave can carry one piece of information
  - Complex waves carry multiple bits of information in a specific order

Visualizing Polymorphism

alpha_mixed

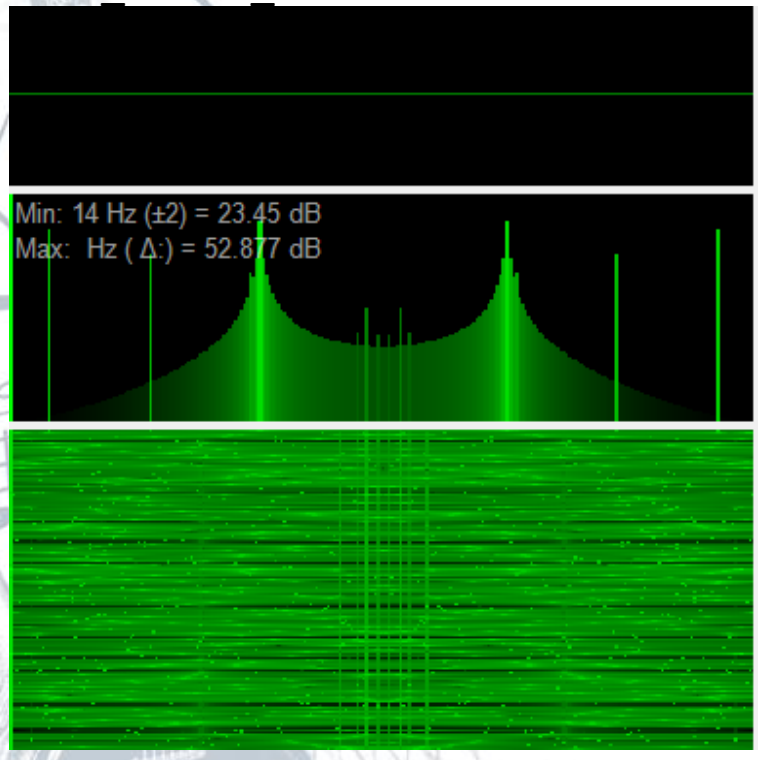


shikata_ga_nai

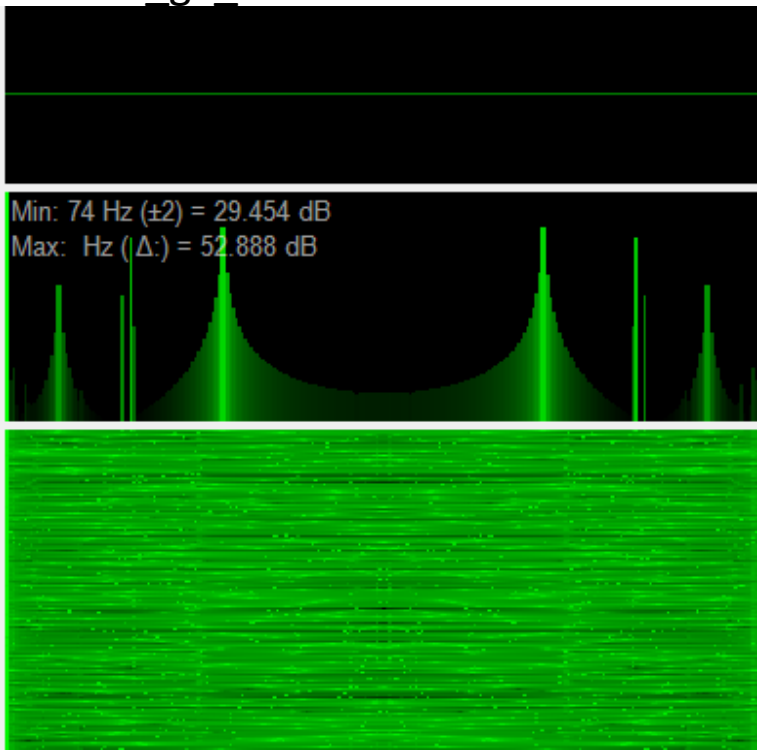


Visualizing Polymorphism

call4_dword_xor



shikata_ga_nai



Tuning In

- Windowing the data allows us to find pure signals...
  - Full packet visualization will not transmit much information
  - Enumerations and type fields become immediately apparent!

Why all the Noise

- This is a proof of concept that may be expanded
- Encoding properties as waves allows magnitude to represent reoccurrence of patterns
- Different attributes such as amplitude can be utilized to represent distance from a target

Thank You

- More info:
  - IIT CS Lectures on You Tube
  - http://rjohnson.uninformed.org