Fourier Transforms and Breaking RSA Encryptions

Nathan Conger

Introduction

Frequencies are everywhere, from the colors you see as light bounces off of them, the signals sent from your phones, and from the music you hear as your favorite artist strums their guitar. How can we make use of frequency? This paper will look into a mathematical transformation known as the Fourier transform. We will start by describing the history of the concept. Afterward, we will derive the transform and look at a simple example. Afterwards, we’ll look into how the transform relates to breaking RSA encryptions using Shor’s algorithm with a quantum computer.

History

The idea of the Fourier Transform dates all the way back to Joseph Fourier in his paper *The Analytical Theory of Heat*. In his paper, Joseph Fourier looked at trying to determine how heat was diffused. In his research, he asserted that any arbitrary continuous function could be represented as a trigonometric series.

Fourier Transform

The Fourier Transform takes in some function over a span of time and decomposes it into its constituent frequencies and amplitudes, such as taking musical chords and finding the frequencies of the notes that compose them. The Fourier Transform is

where is the frequency and is time, is the signal to be decomposed, and is the transformation over the frequency space (Fourier space). Even though the function looks complex, there is a simple way to visualize it. The final term, , relates to counterclockwise rotation around a unit circle with a “winding” frequency . Thus, effectively takes and wind it around the unit circle. If we think of our wound-up function as having mass, then the Fourier transform describes its center of mass in terms of its magnitude. For most winding frequencies the magnitude is small, however there are spikes at each that match an underlying frequency of .

The Fourier Transformation works in both directions, where the Inverse Fourier Transform is

where the input is the Fourier space and the output is some signal . Since the Fourier Transform is reversible, the most obvious application is signal modification. We can pass a signal into the Fourier Transform, apply some sort of modification and invert the modified Fourier Space to create a modified signal. The classic example is removing audio interference from music recording. Suppose you and your band have been recording your next album when unbeknownst to you, your microphone malfunctions. As you play through the recording, you hear a loud high-pitched frequency. With the Fourier Transform, we can isolate the region that irritating frequency appeared and identify its frequency in the Fourier space. With the timeframe and frequency isolated, we can apply some sort of frequency cancelling method, removing it from your recording. Applying the Inverse Fourier to our modified Fourier space object returns the audio with the high-pitched tone removed.

RSA Encryption

This is a cryptosystem where data is securely transferred between two parties via a set of keys that encrypts and decrypts the information. Suppose Bob needs to send sensitive information to Alice. Alice generates a public key, , for prime numbers and and encryption key, , that is smaller and coprime to . Using , , and , Alice generates a private key known only to her. Bob applies N and to his data using an encryption function and sends the encrypted data over to Alice. Alice applies her private key and decrypts the message. The strength of RSA relies on it being extremely difficult to factor large prime numbers. This is because the private key is built using which is generated from the prime numbers.

Generating the Keys and :

* Public key : , where and are large prime numbers.
* Public key exponent : , where
* Private key : Choose D such that

Encryption:

Alice sends Bob and . Bob takes his message M and transforms it into a number by some agreed upon and reversible method. Bob computes the ciphertext with

and sends it over to Alice.

Decryption:

Alice decrypts the ciphertext by applying

onto it. Inverting the message transformation gets her the original message M.

The strength of RSA encryption comes from it being extremely difficult to find the prime factors of numbers. The general number field sieve is the most efficient classical integer factorizing algorithm factors them in exponential runtime.

Shor’s Algorithm & Quantum Fourier Transform

How can the Fourier Transform help with breaking RSA encryptions? We can use the Quantum Fourier Transform to take advantage of properties of quantum mechanics and speed up the process immensely. The runtime of Shor’s algorithm is in polynomial time, leagues faster than the classical approach’s exponential runtime. The algorithm tries to factor the public key *N* takes some terrible guess *g* and spits out a pair of better guesses and that have better odds of being factors of *N*.

Shor’s Algorithm:

1. Choose some random number .
2. Compute , via the Euclidean Algorithm.
3. If , then divides yielding the prime factors ).
4. Otherwise, perform a quantum period finding algorithm to find , the period of the function .
5. If is odd, return to step 1.
6. If , return to step 1.
7. Otherwise, both and are prime factors of *N*.

Period Finding Quantum

Finding the period involves some quantum mechanics. A quantum function simultaneously computes each element in a superposition and outputs a superposition . However, the output state is not known until we measure it. It’s exactly like Schrodinger’s cat, where we don’t know if the cat is alive or dead until we open the box. Typically, direct observation collapses the output into one randomly chosen state. However, if mapping a superposition measures an output that comes from more than one element of the superposition, then the superposition collapses into a superposition just those elements. This property will be significantly useful in determining the period.

The crux of period finding is based on a mathematical fact. If and for some arbitrary integer , then for any integer . Thus, is periodic. It allows us to apply the Quantum Fourier Transform and get the period.

Consider the set of states , where P is an integer and R is the remainder state mod N.

2. Measure the remainder state of the superposition . The superposition collapses into a superposition of the form
3. Apply the Quantum Fourier Transform on the partially collapsed superposition and collapse the result. The collapse results in the final output , some integer .
4. Since collapsing the output yields different results, repeating the process can yield more outputs that can be used to identify the period .

The

References

Miszczak, Jarosław. “Shor's Factoring Algorithm.” *Quantiki*, 25 Oct. 2015, www.quantiki.org/wiki/shors-factoring-algorithm. Publisher is an admin of the Quantiki site. and is a researcher in quantum information at the Institute of Theoretical and Applied Informatics, Polish Academy of Sciences.

Moore, Tristan. “Quantum Computing and Shor's Algorithm.” 7 June 2016.

“RSA Algorithm in Cryptography.” *GeeksforGeeks*, 7 Aug. 2019, www.geeksforgeeks.org/rsa-algorithm-cryptography/.

Struik, Dirk Jan. “Joseph Fourier.” *Encyclopædia Britannica*, Encyclopædia Britannica, Inc., 12 May 2020, www.britannica.com/biography/Joseph-Baron-Fourier.