Cyclone is a symmetric cryptosystem inspired loosely by the famous Enigma machine. The $n \times n$ square matrix key is best thought of as a cylinder of stacked wheels.

Each row or a "wheel" of the key is a permutation on $\{1, 2, 3, ..., n\}$ expressed in the abbreviated form of $f(1)f(2)...f(n-1)f(n)$.

Let f_i be the permutation of row i. Then $f = f_1 \circ f_2 \circ ... f_{n-1} \circ f_n$ is itself a permutation, and it is this f which is used to transform a symbol of the plaintext into a symbol of the ciphertext. Note that this means f is invertible, which is necessary for decoding.

Then this f is shifted to the right in a circular fashion by the value of the plaintext symbol. Finally, each row of the key is itself shifted using this shifted f.

Here's the crucial functions for encoding a single round:

```
function get_f(k)
    f = Int64[]
    n = size(k)[begin]
    for i in 1:n
        x = i
        for j in 1:n x = k[j,x ] end
        push!(f, x)
    end
    f
end
function encode(p,q,F)
    k = copy(q)
    c = Int64[]
    s = zeros(Int64,n)
    for i in eachindex(p)
        f = get_f(k)
        push!(c,f[p[i]] )
        #c[i] = f[p[i]]
        f = circshift(f,p[i])
        push!(F,f)
        for j in 1:n k[j,:] = circshift(k[j,:], f[j]) end
    end
    c
end
```

This is the essence of the algorithm, but the encrypt function uses features like autospin and reversal to make an arbitrary number of rounds (hopefully) more effective.

When $n = 27$, the compositions f tend to be unique. So the key constantly wanders into a new state, meeting each symbol of plaintext with a fresh permutation. I also created a system called Loop which is like a miniature version of Cyclone, using just one row as the entire key.