Glisten is a symmetric cryptosystem that evolved from Cyclone, which was itself inspired loosely by the famous Enigma machine. The key is an $n \times n$ square matrix key, and each row of this matrix is a permutation on $\{1, 2, 3, ..., n\}$, expressed in the abbreviated form of $f(1)f(2)...f(n-1)f(n)$.

Each element $k[i, j]$ represents the response of state i to finding the plaintext j. This response is both the cipher text symbol to be written **and** the next state of the machine. The cipher text is therefore a history of the machine's states.

This might sound insecure, but states as the plaintext is processed. We say that the permutation of row i (which we'll call f_i) is "rolled" by j, when j is added to each of its entries, and the sum is processed by Julia's **mod1** function.

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. This is just modular arithmetic appropriate to Julia's 1-based indexing. In terms of this lingo, f_p is rolled by c after each symbol of plaintext is processed. As expected, p is the current Int64 plaintext symbol, and c is the associated Int64 ciphertext symbol.

This means that the evolution of the key is intensely the function of the plaintext, so that small differences in the plaintext should lead to strong divergence in keystates, hopefully providing more security.

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

```
function encode(p,q)
    k = copy(q)
    c = Int64[]
    m = 1
    for i in eachindex(p)
        push!(c, k[m,p[i]])
        k[p[i],:] = map(x -> mod1(x+c[i],n), k[p[i],:])
        m = c[i]
    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.

I think Glisten, since it uses only fixedsize integers, can be made very fast.