E is a symmetric cryptosystem built on an evolving elementary cellular automata.

A key for the E system is just a typical representation of a CA rule. For instance, a system that uses 2 symbols and a neighborhood of length 5 can be specified with a key of $2^5 = 32$ bits.

A machine is "loaded" with a random key. The simplest, "square" version of a machine uses this key for its initial "row state" too. So the cellular automata is seeded with its own evolution rule.

We "roll" a machine (always "forward") by simply computing its next row. But we want to update the machine's hidden state with bits from the plain text. Here's the essential code, which processes a single symbol of plain text:

```
m.roll()
c << mod(m.r[0] + p[i], m.s)
x := mod(i, m.w)
m.k[x] = mod(m.k[x] + p[i], m.s)
m.r[x] = mod(m.r[x] + p[i], m.s)
```

The machine is rolled forward one row. Then we add the first symbol of its updated row to the plaintext symbol (mod the number of symbols) to get the ciphertext symbol. Then we add the plaintext symbol to both the key and the row (mod the number of symbols) at position i (mod the length of the row / key). We end up with a helix of insertions of plaintext bits (though insertions of 0 have no effect.) The key (the evolution rule) is constantly updated, and the row state is also constantly updated. The goal is to entangle hidden state with the plaintext **and** to leverage the presumed randomness of the plaintext so that simple loops are avoided in the hidden state. Cellular automata sometimes fall into simple loops, especially with very small keys.

While E is primarily a sculpture (created for aesthetic pleasure), it was created within the constraint that it be a genuine cryptosystem. I'm not a specialist in the field, but I think that it's a decent system (?) if one picks and tests a good key. I imagine that keys of all 1s or all 0s might be terrible, but exploring random keys suggests that most keys of 32 bits or more a good.

I should mention that E includes rounds and the reversal of texts between rounds as well as an autospin feature. I think these features are easy enough to make sense of from the code. The idea of “autospin” is the meet each round with a fresh “key” that is of course determined by the actual key.

What’s presented as E is just a variant of many possible, similar systems. The goal is as much simplicity and beauty that one can get away with it without the system being terrible. So I don’t add messy twists and turns unless they (seem to) make the system stronger. I also assume that anyone interested in E will likely have the skills and motivation to make their own variants. I’d be glad to discuss CA-based crypto with others, too.