

Welcome !

This document summarizes featured projects and provides links to more information, including source code.

Cryptography

THORIUM / CESIUM

This symmetric cryptosystem uses a square matrix of symbols, such as bits, as a key. The rows and columns are “circularly shifted,” so that the matrix is topologically a torus. The diagonal of the matrix and its trace plays a central row.

<https://k0ntinuum.github.io/th0rium/>

CONE

This symmetric cryptosystem is based on a “triangular” implementation of ternary elementary cellular automata. The key is 27 ternary symbols which represent a rule for a ternary cellular automaton with a neighborhood size of 3. The rule is then also used as a seed (along with a symbol of plaintext) in order to get the ciphertext symbol.

<https://k0ntinuum.github.io/c0ne/>

FORTEX

This system, like Cone, features a triangular key. Like Thorium.Cesium, the base used is adjustable. Rows of the triangle are rotated, and a central column functions like an escalator (all shifts are circular). The modular sum of elements in this central column is added to plaintext symbols, and the apex of the pyramid is used to represent this changing value in console “textgraphics” demonstrations.

<https://k0ntinuum.github.io/f0rtex/>

PRE

Pre or Prefix is a symmetric cryptosystem featuring prefix or instantaneous codes. One input symbol might become several output symbols, and, one output symbol might condense several input symbols. Each state of the machine writes a prefix code, which allows for decoding, but the states together write a union of prefix codes which is not in general itself a prefix codes. This means that the ciphertext is difficult to tokenize.

<https://k0ntinuum.github.io/prEfix>

Elementary and Continuous Cellular Automata

FFLO

Fflo implements a continuous cellular automaton, allowing (in most versions) the user to *play* the automaton like a visual-musical instrument. Of all the programs featured here, Fflo is the most graphics intensive.

<https://k0ntinuum.github.io/ffl0>

STARSHIP

This program (really family of programs) provides a live visual exploration of the computational space of ECA or elementary cellular automata. Wolfram focused (at least most famously) on binary automata, but Starship generalizes this to an arbitrary base. This program can also be used to generate “quasi-organic” abstract art.

<https://github.com/k0ntinuum/starship>

Artificial Neural Networks

SCOPE

This program visualizes the interactive real-time learning of an artificial neural network. The user can vary the architecture of the network, adjust the learning rate, reset parameters, and so on. I wrote the first version in C while studying deep learning in grad school. Other versions followed in Rust, Go, and Javascript. The JS browser version (<https://k0ntinuum.github.io/scope-js/>) is extremely simplified (the user can choose between 3 buttons).