Implementation of a Cellular Automaton for Hard Symmetric Cryptography

Computing capability is expanding at an exponential rate, and what was secure 5 years ago will not be secure 5 years from now. Current methods for encryption commonly use the factors of large numbers to generate keys, an asymmetric approach which requires two cryptographic keys but relies on the difficulty of finding the prime factors of large numbers. The authors propose a new approach that uses a cellular automaton to encrypt and decrypt the message. This could potentially future-proof cryptographic algorithms for a time when computers will be able to find solutions to factoring problems, as certain classes of cellular automata have been proven to be sufficiently complex as to have no mathematical simplification. Messages are encrypted by running a cellular automaton seeded from the cryptographic key while performing sequential bitwise XOR operations. Decryption takes the seed used in the encryption, creates the cellular automaton with the proper number of generations, and works backwards from the encrypted message. Sets of 15,000 messages were generated in python, from which the frequency and value change of the messages were analyzed. Results show that the specific implementation used is vulnerable to analysis of character frequency and predictable variation of character displacement values. Additional work will need to be performed to verify which features of the implementation result in the security vulnerabilities, as well as to conclusively determine whether or not a cellular automaton is a viable method of hard cryptographic encoding.