Cryptography with Cellular Automata

Stephen Wolfram
The Institute for Advanced Study, Princeton NJ 08540.

(November 1985)

EXTENDED ABSTRACT*

This abstract discusses a stream cipher based on a simple one-dimensional cellular automaton. The cellular automaton consists of a circular register with N cells, each having a value a_i equal to 0 or 1. The values are updated synchronously in discrete time steps according to the rule

$$a_i' = a_{i-1} \text{ XOR } (a_i \text{ OR } a_{i+1}) \ , \tag{1a}$$

or, equivalently,

$$a_i' = (a_{i-1} + a_i + a_{i+1} + a_i a_{i+1}) \bmod 2 \ . \tag{1b}$$

The initial state of the register is used as a seed or key. The values $a^{(i)}$ attained by a particular cell through time can then serve as a random sequence. Ciphertext C can be obtained from binary plaintext P as usual according to $C_i = P_i \text{ XOR } a^{(i)}$; the plaintext can be recovered by repeating the same operation, but only if the sequence $a^{(i)}$ is known.

Cellular automata such as (1) have been investigated in studies of the origins of randomness in physical systems [2]. They are related to non-linear feedback shift registers, but have slightly different boundary conditions.

Figure 1 shows the pattern of cell values produced by (1) with a seed consisting of a single nonzero cell in a large register. The time sequence of values of the centre cell shows no statistical regularities under the tests of ref. [3] (for sequence lengths up to $2^{19} \equiv 5 \times 10^5$). Some definite spacetime patterns are nevertheless produced by the cellular automaton rule.

In the limit $N \to \infty$, the cellular automaton evolution is like an iterated continuous mapping of the Cantor set, and can be studied using dynamical systems theory [4]. One result is that the evolution is unstable with respect to small perturbations in the initial seed. A change produced by reversing a single cell value typically expands at a rate given by Lyapunov exponents, equal to 0.25 on the left, and 1

* Many more details are given in ref. [1].

H.C. Williams (Ed.): Advances in Cryptology - CRYPTO '85, LNCS 218, pp. 429-432, 1986.

on the right. Length T time sequences of cell values are found however to be affected on average only by about $1.19T$ initial values.

Iterations of the cellular automaton rule (1) can be considered as Boolean functions of initial cell values. Disjunctive normal forms (minimized using [5]) for these functions are found to increase in size roughly as $4^{0.65t}$, giving some indication of the complexity of the cellular automaton evolution.

Figure 2 shows the complete state transition diagram for the cellular automaton (1) in a register of size N=11. For large N, an overwhelming fraction of states lie on the longest cycle. But there are also shorter cycles, often corresponding to states with special symmetries. Figure 3 shows the length of the longest cycle as a function of N. The results (up to N=53, which gives cycle length 40114679273) fit approximately $2^{0.61N}$. The mapping (1) is not a bijection, but is almost so; only a fraction $(\kappa/2)^N \cong 0.85^N$ of states do not have unique predecessors [6] (κ is the real root of $4\kappa^3-2\kappa^2-1=0$).

The security of a cryptographic system based on (1) relies on the difficulty of finding the seed from a time sequence of cell values. This problem is in the class NP. No systematic algorithm for its solution is currently known that takes a time less than exponential in N. No statistical regularities have been found in sequences shorter than the cycle length.

One approach to the problem of finding the seed [6] uses the near linearity of the rule (1). Equation (1) can be written in the alternative form $a_{i-1} = a_i'$ XOR (a_i OR a_{i+1}). Given the values of cells in two adjacent columns, this allows the values of all cells in a triangle to the left to be reconstructed. But the sequence provided gives only one column. Values in the other column can be guessed, and then determined from the consistency of Boolean equations for the seed. But in disjunctive normal form the number of terms in these equations increases linearly with N, presumably making their solution take a time more than polynomial in N.

The cellular automaton (1) can be implemented efficiently on an integrated circuit; it requires less than ten gate delay times to generate each output bit, and can thus potentially be used in a variety of high-bandwidth cryptographic applications.

Much of the work summarized here was done while I was consulting at Thinking Machines Corporation (Cambridge, MA). I am grateful for discussions with many people, including Persi Diaconis, Carl Feynman, Richard Feynman, Shafi Goldwasser, Erica Jen and John Milnor.

References

1. S. Wolfram, "Random sequence generation by cellular automata", to be published in *Advances in Applied Mathematics*.
2. S. Wolfram, "Origins of randomness in physical systems", Phys. Rev. Lett. **55**, 449 (1985); S. Wolfram, "Cellular automata as models of complexity", Nature **311**, 419 (1984).
3. D. Knuth, *Seminumerical Algorithms*, (Addison-Wesley, 1981).
4. S. Wolfram, "Universality and complexity in cellular automata", Physica **10D**, 1 (1984).
5. R. Rudell, *espresso* software program, Computer Science Dept., University of California, Berkeley (1985).
6. C. Feynman and R. Feynman, private communication.

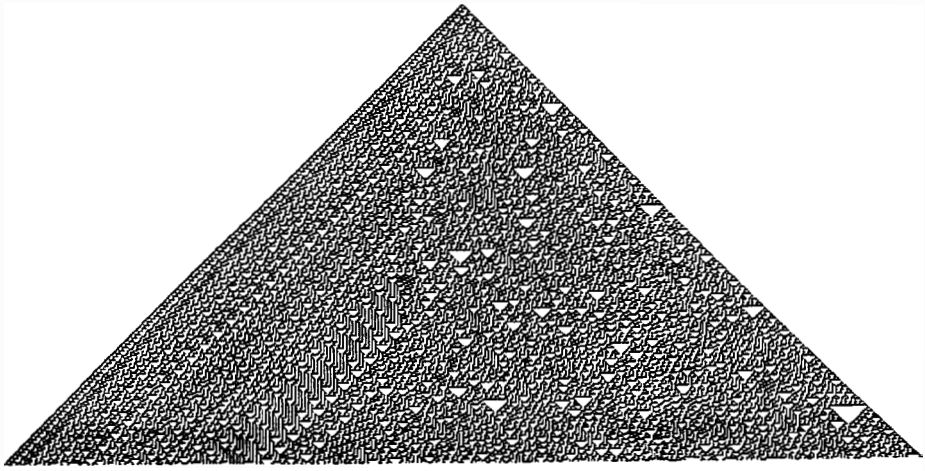

Figure 1. Pattern produced by evolution according the cellular automaton of eqn. (1) from a simple seed containing a single nonzero bit. 250 successive states of an arbitrarily large register are shown; black squares represent nonzero cells. Columns of cell values, say in the centre, seem random for practical purposes.

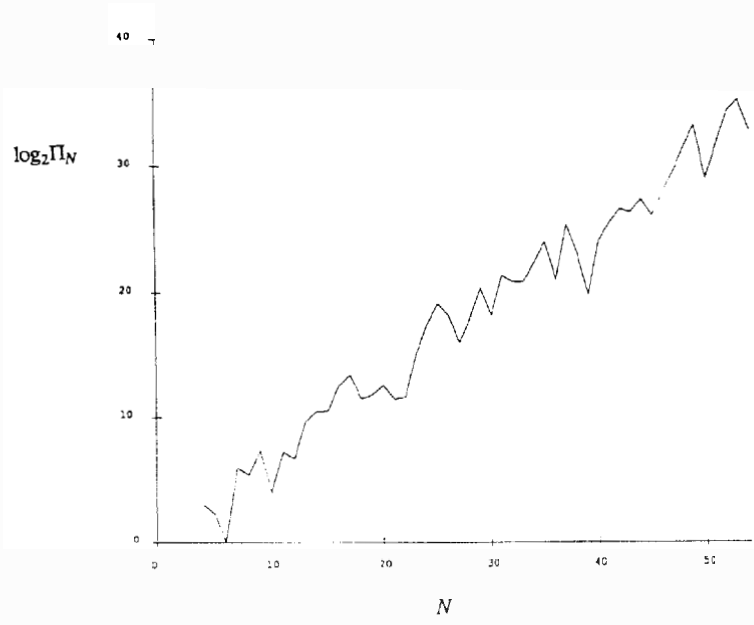

Figure 3. Length Π_N of the longest cycle as a function of register size N.

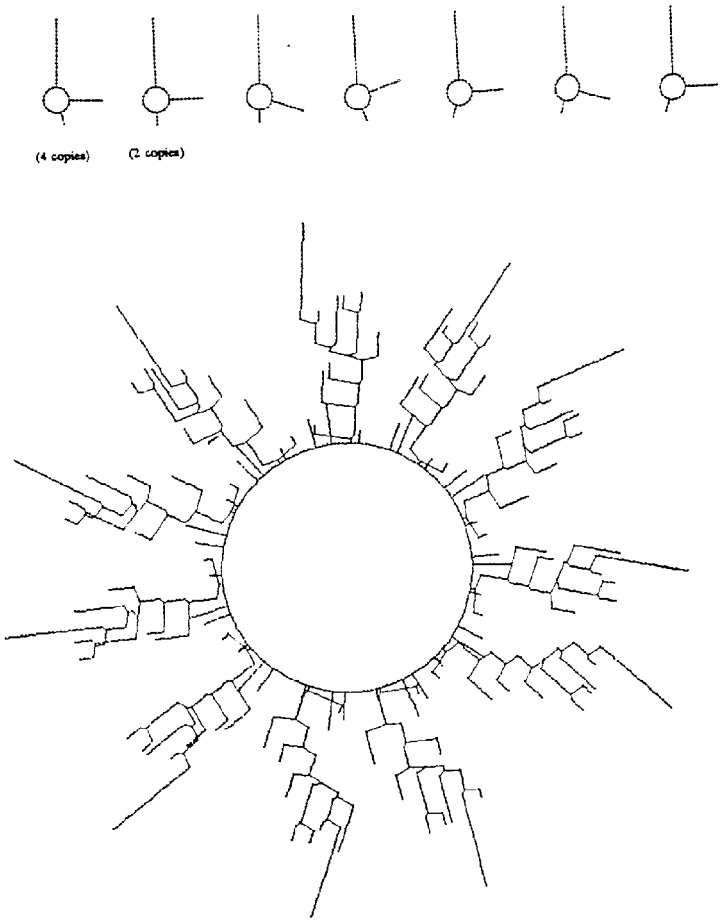

Figure 2. Complete state transition diagram for the cellular automaton of eqn. (1) in a circular register of size N=11. There are 2^N states, each represented by dots. Evolution from any state leads eventually to one of the cycles shown.