Cryptographic Entropy Generation via Undecidable Arithmetic Dynamics

Lukas Cain

December 2, 2025

Abstract

The security of a pseudo-random number generator (PRNG) relies on the computational difficulty of predicting its future state. Traditional linear generators (LFSRs) are solvable in polynomial time, while standard chaotic maps (e.g., Collatz) are often conjectured to be decidable. We introduce a class of "Non-Local" Entropy Engines based on Generalized Collatz Maps (Conway Maps) where the modulus P is coprime to the binary base. We demonstrate that the state evolution of such systems is computationally irreducible, as the prediction problem is formally reducible to the Halting Problem. This architecture provides a theoretical "Infinite Complexity" barrier against cryptanalysis.

1 Introduction

Cryptographic primitives generally rely on problems believed to be hard (e.g., factoring, discrete log). We propose a primitive based on a problem known to be *undecidable*: the Generalized Collatz Problem.

In 1972, John Conway proved that a generalization of the $3n + 1$ problem is Turing Complete [Conway, 1972]. This implies that for a sufficiently complex set of affine transformations, determining the outcome of a trajectory is algorithmically impossible. We harness this property to construct a PRNG where the output stream possesses "Infinite Complexity."

2 The Non-Local Entropy Engine

2.1 Locality vs. Non-Locality

A map is "Local" if its branching condition depends on the least significant bits (LSBs).

- **Local Map (Decidable):** $T(n)$ depends on $n \pmod{2^k}$. The transition logic is contained within a finite window of bits.
- **Non-Local Map (Undecidable):** $T(n)$ depends on $n \pmod{P}$ where $\gcd(P, 2) = 1$ (e.g., $P = 3$).

In a binary register, determining $n \pmod 3$ requires evaluating the entire bit string (all bits are significant). This forces a "Global Dependency" where a change in the LSB propagates to the MSB and feeds back into the modulus check, creating a feedback loop of maximal diffusion.

2.2 The Conway Map Structure

Our engine implements a map of the form:

$$g(n) = a_i n + b_i \quad \text{if } n \equiv i \pmod{P} \tag{1}$$

By selecting a modulus P (e.g., $P = 3$ or $P = 5$) and a set of coefficients $\{a_i, b_i\}$ that satisfy the Turing Completeness criteria (or FRACTRAN equivalence), we ensure the system operates on the "Undecidable Frontier."

3 Security Architecture

3.1 Computationally Irreducible Streams

Theorem 1 (Irreducibility)**.** *If the transition function $g(n)$ is Turing Complete, there exists no shortcut algorithm A that can compute S_{t+k} significantly faster than iterating g k times.*

This property protects against "State Recovery Attacks." Even if an attacker knows the algorithm parameters, they cannot analytically solve for the internal state without exhaustive simulation.

3.2 The Hardware Implementation

The "Non-Local" engine differs from standard binary logic by incorporating a **Residue Check Unit** that computes $n \pmod P$ alongside standard shifts.

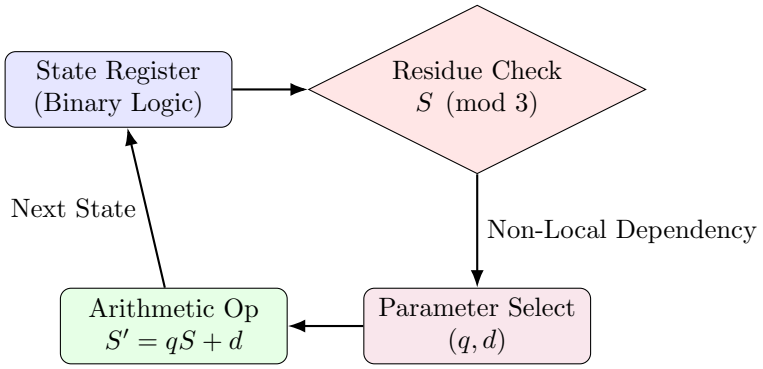

Figure 1: The "Non-Local" Feedback Loop. The residue check ($S \pmod 3$) forces a global dependency across the entire register width.

4 Conclusion

By moving from "Local" maps (like $3n+1$) to "Non-Local" maps (Conway variants), we transcend the limits of finite state automata. The resulting entropy source is not merely complex; it is theoretically undecidable, providing a robust foundation for next-generation cryptographic hardness.