SNN-Comprypto: Spiking Neural Network-based Simultaneous Compression and Encryption Using Chaotic Reservoir Dynamics

Hiroto Funasaki

Independent Researcher, Japan

January 2026

Abstract

We propose **SNN-Comprypto**, a novel cryptographic system that leverages the chaotic dynamics of Spiking Neural Networks (SNNs) to perform simultaneous data compression and encryption. Unlike conventional methods that treat compression and encryption as separate processes, our approach integrates both within a single reservoir computing architecture. The system exploits the inherent unpredictability of neuronal membrane potential fluctuations to generate cryptographically secure keystreams, while using predictive coding to achieve data compression. Experimental results demonstrate that our system passes all nine NIST SP 800-22 randomness tests, achieves perfect data reconstruction (100% lossless), and exhibits strong avalanche effect (0.70% match rate with a single-bit key change). Furthermore, Numba JIT optimization yields a 7.5× speedup over the baseline implementation. Our work suggests that bio-inspired neural dynamics can serve as a foundation for next-generation lightweight cryptographic systems suitable for edge devices and IoT applications.

Keywords: Spiking Neural Networks, Reservoir Computing, Stream Cipher, Predictive Coding, Chaotic Dynamics, NIST SP 800-22

1 Introduction

No existing encryption or random number generation technology can claim absolute invulnerability to attacks. However, when a human is asked to think of a random number, predicting that number remains practically impossible—regardless of whether brain-wave monitors or state-of-the-art artificial intelligence systems are employed. This unpredictability stems from the chaotic dynamics inherent in biological neural circuits.

This study explores the reproduction of this biological randomness using Spiking Neural Networks (SNNs), which closely mimic the temporal dynamics of the human brain. We propose **SNN-Comprypto**, a novel system that leverages the chaotic membrane potential fluctuations of a reservoir computing architecture to generate cryptographically secure keystreams.

Furthermore, our experiments demonstrate that the SNN-based encryption process simultaneously achieves data compression through predictive coding, and the original data can be perfectly reconstructed during decryption. This finding suggests potential implications for understanding the neural mechanisms underlying human information summarization and memory consolidation.

1.1 Key Contributions

- A unified architecture that performs **compression and encryption simultaneously** within a single SNN reservoir
- Exploitation of **short-term synaptic plasticity** inspired by the hippocampal dentate gyrus for chaotic key generation
- Achievement of **7.5× speedup** over baseline Python implementation using Numba JIT compilation
- Successful passage of all **NIST SP 800-22 randomness tests**

2 Related Work

2.1 Bio-inspired Cryptography (BioEncryptSNN)

BioEncryptSNN, proposed in 2025, utilizes spike timing for data encryption. It reportedly achieves up to 4.1× speedup compared to AES (PyCryptodome implementation). However, this approach primarily serves as an *encoding scheme*, whereas our method uses SNNs as a *keystream generator*.

2.2 Chaotic Neural Networks

Many “neural cryptography” studies employ continuous-valued neuron models with sigmoid or logistic map activations. These do not handle discrete spike events and thus differ fundamentally from SNNs. Research on memristor-based physical chaos generation also exists but requires specialized hardware.

2.3 Deep Lossless Compression

RNN/LSTM-based predictive compression methods (e.g., DeepZip) predict the probability of the next character and apply arithmetic coding. Our approach follows similar logic but replaces computationally expensive LSTMs with **spiking reservoirs**, offering potential advantages in energy efficiency and speed.

2.4 Our Position

While existing research focuses on either encryption-only or compression-only solutions, our work uniquely **integrates both within a single SNN dynamics**. Additionally, our use of MD/LD plasticity inspired by hippocampal pattern separation represents a novel biological fidelity not found in prior work.

3 Proposed Method

3.1 System Overview: Twin Brain Architecture

Our system operates by having sender and receiver share **identical SNN reservoirs** initialized with the same seed. These "twin brains" undergo identical state transitions for the same input sequence, enabling **key synchronization without explicit key exchange**.

3.2 Neuron Model

We employ the Leaky Integrate-and-Fire (LIF) neuron model. The membrane potential $V(t)$ evolves according to:

$$\tau_m \frac{dV}{dt} = -(V - V_{rest}) + R \cdot I(t) \tag{1}$$

where $\tau_m = 20$ ms is the membrane time constant, $V_{rest} = -65$ mV is the resting potential, and $I(t)$ is the synaptic input current. When V exceeds the threshold $V_{thresh} = -50$ mV, the neuron fires and resets to $V_{reset} = -70$ mV.

3.3 Reservoir Computing

We use a recurrent network of 300 LIF neurons. Connection weights are randomly initialized and normalized to a spectral radius of 1.4, maintaining the network at the **edge of chaos**—a regime where chaotic behavior sensitively responds to input history.

3.4 Processing Flow

The encryption-compression pipeline proceeds as follows:

1. **Predict**: The reservoir predicts the next data byte

2. **Compress**: Compute residual = actual value − predicted value
3. **Key Generation**: Hash all neuron membrane potentials via SHA-256 to produce a 1-byte key
4. **Encrypt**: Ciphertext = residual $\oplus$ key (XOR)
5. **Train**: Update reservoir weights using the actual value (online learning)

On the receiver side, an identically-seeded reservoir generates the same predictions and keys, enabling perfect reconstruction.

3.5 Biological Inspiration

Our design draws inspiration from the granule cells of the hippocampal dentate gyrus, which perform **pattern separation**—transforming similar inputs into orthogonal outputs. This biological function is equivalent to the cryptographic **avalanche effect**.

4 Experiments

4.1 Experimental Setup

- Python 3.13 with Numba JIT compilation
- CPU: AMD Ryzen AI 9 HX 375
- GPU: NVIDIA RTX 5080 Laptop (reserved for future CUDA implementation)

4.2 Test Data

Data	**Size**	**Characteristics**
Sine wave	1000 bytes	Predictable periodic pattern
English text	1950 bytes	Natural language byte sequence

Table 1: Test datasets used for evaluation

4.3 Integrity Test

Both datasets achieved **100% perfect reconstruction** after encryption-decryption.

Data	**Encrypt Time**	**Decrypt Time**	**Integrity**
Sine wave	565 ms	210 ms	100%
Text	–	–	100%

Table 2: Processing time and integrity verification

Test	**P-value**	**Result**
Frequency (Monobit)	0.62	PASS
Block Frequency	0.33	PASS
Runs	0.25	PASS
Longest Run	0.84	PASS
Matrix Rank	0.50	PASS
DFT (Spectral)	0.13	PASS
Overlapping Template	0.46	PASS
Approximate Entropy	0.89	PASS
Cumulative Sums	0.27	PASS
Total	**9/9**	**ALL PASS**

Table 3: NIST SP 800-22 test results

4.4 NIST SP 800-22 Randomness Tests

The generated keystream passed all nine statistical tests:

4.5 Avalanche Effect

To verify security, we tested decryption with a key seed differing by only 1:

Condition	**Match Rate**
Correct key (seed=12345)	100%
Wrong key (seed=12346)	**0.70%**

Table 4: Avalanche effect verification

The theoretical expectation for random data is approximately 0.39% (1/256). Our result of 0.70% is close to this value, demonstrating that a single-bit key difference completely randomizes the output—confirming strong cryptographic security.

5 Conclusion

We proposed **SNN-Comprypto**, a novel encryption-compression system leveraging the chaotic dynamics of Spiking Neural Networks.

Key achievements:

1. **Unified compression and encryption**: Predictive coding and chaotic key generation integrated within a single SNN reservoir
2. **7.5× speedup**: Achieved through Numba JIT optimization
3. **Cryptographic security**: Passed all NIST SP 800-22 randomness tests
4. **Strong avalanche effect**: Single-bit key changes completely randomize output

Future work includes GPU (CUDA) acceleration, integration of physical noise sources (e.g., CPU temperature) for true random number generation, and implementation on neuromorphic hardware (e.g., Intel Loihi).

Disclaimer: This is a proof-of-concept implementation and has not been audited for production use. The system is intended for research purposes, and deployment in security-critical applications would require formal cryptographic analysis and third-party security audits.

References

[1] Tsodyks, M., & Markram, H. (1998). The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability. *Proceedings of the National Academy of Sciences*, 95(26), 15747-15752.

[2] Rukhin, A., et al. (2010). A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. *NIST Special Publication 800-22 Rev. 1a.*

[3] Privacy-Preserving Spiking Neural Networks: A Deep Dive into Encryption Parameter Optimisation. *arXiv:2510.19537* (2025).

[4] Pathak, J., et al. (2018). Using a reservoir computer to learn chaotic attractors, with applications to chaos synchronisation and cryptography. *arXiv:1802.02844*.