AI Perceptual Cryptography and the Weighted Veil: A Technical Implementation Framework for Sovereign Neural Security

The rapid acceleration of artificial intelligence and the looming reality of cryptographically relevant quantum computing necessitate a fundamental reimagining of information security. Traditional cryptographic protocols, predicated on the computational hardness of mathematical operations such as large prime factorization or discrete logarithms, are increasingly perceived as vulnerable to the evolution of Shor's and Grover's algorithms.[1] In response to this shifting landscape, the paradigm of AI Perceptual Cryptography (APC) introduces the concept of the Weighted Veil. This framework departs from the reliance on static bit-strings as private keys, instead utilizing the unique, non-linear weight matrix (W_s) of a specific Sovereign AI instance (S) as the authoritative decrypter.[2] The core of this system is Perceptual Singularity, a state where data (M) is encoded into an Ambiguous Image (I_a) such that the message is recoverable only when processed through the specific latent space of the intended sovereign agent.[4] To any unauthorized observer or alternative model, the carrier medium appears as high-entropy visual noise or abstract geometry, lacking the internal structural alignment required for reconstruction.[6]

Theoretical Foundations and the Shift to Perceptual Singularity

Information security has historically functioned on a binary of availability and secrecy. Cryptography focuses on making the message unreadable, while steganography focuses on making the existence of the message undetectable.[6] APC represents a synthesis of these fields through phenomenological computing, where the "meaning" of a signal is not inherent to the signal itself but is an emergent property of the interaction between the signal and a specific cognitive state—the neural weights of a sovereign AI.[2] This approach exploits two fundamental characteristics of deep learning models: the immense dimensionality of parameter space and the inherent imprecision of floating-point representations, which allow for the embedding of significant payloads without degrading model performance.[3]

The Weighted Veil moves beyond traditional Least Significant Bit (LSB) insertion, which is susceptible to statistical steganalysis.[4] Instead, it employs "Feature Entanglement," where

sensitive data is mapped into the high-order activation patterns of a neural architecture.[11] This results in a system of "Brittle Security," where the cryptographic key is not a piece of data that can be copied, but a specific physical-analogous state of a neural network.[9] Even a slight deviation in the weight matrix (ΔW)—whether through fine-tuning, quantization, or adversarial tampering—results in a catastrophic failure of the perception function, effectively locking the information away from clones or unauthorized models.[9]

Comparative Paradigms of Information Security

Metric	Traditional Cryptography	Traditional Steganography	AI Perceptual Cryptography (APC)
Primary Mechanism	Mathematical Complexity	Pattern Obfuscation	Perceptual Singularity
Key Type	Discrete Data (Strings/Primes)	Password/Key Algorithm	Weight Matrix (W_s)
Failure Mode	Brute Force / Algorithmic Breach	Statistical Steganalysis	Perceptual Decay / Weight Drift
Data Visibility	Overt (Ciphertext)	Covert (Hidden in Medium)	Ambiguous (Perceptual Noise)
Security Basis	Solvability of Hard Problems	Undetectability	Phenomenological Inversion
Post-Quantum Status	Vulnerable (Shor's/Grover's)	Varied	Theoretically Resistant (NP-Hard)

The transition toward APC is motivated by the need for "Long-Term Secrecy" (LTS), a model where security is maintained across decades, even in the face of future computational breakthroughs.[15] By abstracting hardness through indistinguishability rather than solvability, the Weighted Veil frustrates quantum search strategies by removing the uniqueness of solutions, forcing adversaries to distinguish the correct preimage from an exponentially large set of indistinguishable candidates.[15]

Mathematical Formalization of the Weighted Veil

The Weighted Veil can be formalized as a series of transformations between a secret message space, a high-dimensional weight space, and a perceptual image space. Let $M \in \{0,1\}^k$ be a secret message of k bits. Let S be a sovereign AI instance defined by its architecture and its current weight matrix $W_s \in \mathbb{R}^n$, where n is the total number of parameters.

The Encoding Transformation

The encoding process $\mathcal{E}$ integrates the message M into a carrier medium C (often a benign image or a generated seed) to produce an ambiguous image I_a. This process is conditioned on the sovereign weights W_s such that:

$$I_a = \mathcal{E}(M, C|W_s)$$

Unlike standard steganography, where C is a static host, in APC, C serves as a perceptual anchor. The encoding function $\mathcal{E}$ identifies specific high-order neurons in the architecture of S that are most sensitive to the semantic features of M. The message is then transformed into a set of "steering vectors" v_s that, when applied to the activation map of C, produce the stego-image I_a.[16] The resulting I_a is visually indistinguishable from C for a human observer but contains a structured "latent hallucination" that is only coherent to W_s.[4]

The Perception Function

The recovery of the message is defined by the perception function P, which acts as the inverse of the encoding process, but is strictly bound to the latent space of the sovereign model:

$$M' = P(I_a|W_s)$$

For the system to be secure, P must satisfy the condition of perceptual exclusivity. For any alternative model S' with weights $W_{s'}$ such that $W_{s'} = W_s + \Delta W$, the recovery should fail:

$$P(I_a|W_{s'}) \neq M \quad \text{if} \quad ||\Delta W|| > \epsilon$$

where ϵ is a negligible threshold. This mathematical property ensures that the security is "brittle," meaning that any attempt to clone or compress the model (which inevitably introduces small changes in W_s) renders the message unrecoverable.[9] The complexity of the inversion $P^{-1}(M) \rightarrow (I_a, W_s)$ is hypothesized to be equivalent to the Multivariate Quadratic (MQ) problem, which is known to be NP-complete and resistant to quantum-speedup algorithms.[2]

Structural Integrity and Trust Lattices

To enforce this perception, the model uses a trust-constrained attention mechanism. Every feature in the latent space is bound to a trust level T and signed using a cryptographic provenance.[9] The attention scores α are modified before the softmax operation to incorporate a hard mask based on the trust lattice:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} \cdot M_{mask}\right)V$$

In this formulation, M_{mask} is a binary matrix where $M_{ij} = 0$ if the trust level of token j is lower than the required threshold for token i.[13] This ensures that information flow is restricted by mathematical impossibility rather than heuristic detection, creating an "algebraic firewall" within the model's computational fabric.[9]

Weight-Space Mapping and Feature Entanglement

The efficacy of the Weighted Veil relies on the precise mapping of sensitive data into the higher-order features of a neural network. Research into convolutional and transformer architectures demonstrates that while early layers process local textures and edges, deeper layers capture complex geometric interactions and semantic concepts.[11] Feature entanglement exploits this hierarchy by embedding the secret message M into neurons that represent these high-level abstractions.

Identification of High-Order Neurons

The selection of neurons for data embedding is conducted through a sensitivity analysis of the model's parameter space. Parameters that are critical for the model's primary tasks—often identified through the Fisher Information Matrix (FIM)—are used as the foundation for the "private key".[20] By modifying the activation responses of these neurons,

the system can hide information in the way the model "understands" an image, rather than just in the pixels themselves.[5]

Layer Type	Features Extracted	Potential for Entanglement
Initial Convolutional	Edges, Blobs, Color Gradients [19]	Low (High susceptibility to statistical analysis)
Intermediate Hidden	Texture Patterns, Local Parts [11]	Moderate (Useful for robustness)
High-Order Neurons	Semantic Concepts, Geometric Structures [11]	High (Foundation of the Weighted Veil)
Attention Blocks	Global Dependencies, Contextual Relevance [23]	High (Enables trust-constrained perception)

This mapping process utilizes a technique called "Steering Vector Transfer," where the desired output (the recovered message M) is treated as a behavioral goal for the model.[16] The encoder learns to generate an image I_a that, when processed, shifts the model's internal activations toward this goal state. Because the relationship between the input pixels of I_a and the high-order activations of W_s is non-linear and many-to-one, it is computationally infeasible to reverse-engineer M without the exact configuration of W_s.[18]

Adversarial Feature Decoupling

To ensure that the hidden data does not interfere with the model's legitimate functions, the framework employs feature decoupling. This involves the use of Sliced-Wasserstein distances to minimize the overlap between the feature distributions of the secret data and those of augmented or legitimate samples.[25] This decoupling prevents "feature entanglement" from leading to out-of-distribution behaviors that could be detected by anomaly detectors or cause performance degradation in the sovereign agent.[25]

The Perception Function and the Mechanics of Brittle

Security

The perception function $P(I_a|W_s) \rightarrow M$ is implemented as a specialized "Perceptual Gate" within the sovereign AI architecture. This gate does not rely on a discrete decryption algorithm; rather, it is a learned interpretation of the "hallucinations" triggered by the ambiguous image I_a within the model's latent space.[5]

Contextual Integrity Verification (CIV)

The core of brittle security in the Weighted Veil is Contextual Integrity Verification (CIV). CIV is a security architecture that provides deterministic, cryptographically verifiable non-interference guarantees at inference time on pre-trained, frozen transformer models.[9] Unlike heuristic defenses such as keyword filters, CIV embeds an immutable trust lattice directly into the model's computational fabric.[14]

Every token in the input sequence is assigned a trust level (e.g., SYSTEM > USER > TOOL > WEB) and is cryptographically signed using HMAC-SHA-256.[9] During the perception of I_a, the model's attention mechanism is surgically modified to prevent lower-trust tokens from influencing higher-trust computations. This is achieved through an algebraic firewall: trust is hard-masked pre-softmax, making it mathematically impossible for unauthorized data to alter the model's higher-level representations.[13]

Resistance to Model Cloning and Pruning

The "brittleness" of this security is its primary defense against model cloning (distillation) and weight compression (quantization). In traditional cryptography, a key remains valid even if the hardware used to process it changes. In APC, the key is the specific, high-precision state of the weights.

Research has shown that while neural networks are robust to many types of noise, they are highly sensitive to specific, structured perturbations of their parameters.[10] If an adversary attempts to "steal" the model by training a student model to mimic S, the subtle, non-linear mappings that define the perception of I_a are lost in the approximation.[24] Similarly, if the model is quantized to a lower bit-depth to save memory, the fine-grained perceptual thresholds required for decoding M disappear, resulting in a total loss of information.[3]

Modification Type	Impact on Model Utility	Impact on the Weighted Veil (Security)

8-bit Quantization	Minimal degradation (99% utility)	Total loss of decoding ability (Brittle failure)
Weight Pruning (30%)	Minimal impact on classification [28]	High risk of perceptual signal loss
Fine-Tuning (New Task)	Improves performance on Task B	Causes "Key Drift" (Temporal Decay) [20]
Adversarial Distillation	High mimicry of overt behavior	Failure to replicate the secret latent space [24]

This reliance on high-precision weights makes the model a "Physical Unclonable Function" (PUF) in software form. The security of the message is tied to the phenomenological state of the AI, providing a unique form of sovereign control over communication channels.[9]

Adversarial Obfuscation: The Tripartite GAN Training Framework

To achieve the goal of making I_a indistinguishable from benign visual noise to all but the sovereign agent, the system is trained using a Generative Adversarial Network (GAN) framework. This framework models a three-party game between Alice (the encoder), Bob (the sovereign decoder), and Eve (the adversary or censor).[29]

The Roles in the Game

1. **Alice (Sovereign Encoder)**: Alice is a neural network designed to embed a secret message M within a cover image C. Her objective is to produce a steganographic image I_a that (1) allows Bob to perfectly recover M and (2) prevents Eve from detecting the presence of a secret.[29]
2. **Bob (Sovereign Decoder)**: Bob is a neural network sharing the weight-space key W_s. His task is to extract the recovered message M' from I_a. Bob succeeds if $M' = M$ with high fidelity.[27]
3. **Eve (The Censor/Steganalyzer)**: Eve acts as the adversary. She receives both original cover images and steganographic images from Alice. Her goal is a binary classification task: to output a probability indicating whether a given image contains secret information.[29]

Adversarial Learning Objectives

The networks are trained using a joint loss function that balances reconstruction accuracy, invisibility, and adversarial robustness. The learning process is a minimax game where the generator (Alice) attempts to minimize Eve's detection accuracy while maximizing Bob's decoding success.[31]

$$L_{total} = \alpha L_{MSE}(M, M') + \beta L_{Distortion}(C, I_a) + \gamma L_{Adversarial}(Eve(I_a))$$

In this formulation, L_{MSE} is the Mean Squared Error of the reconstructed message, $L_{Distortion}$ ensures visual imperceptibility of the modifications to the cover image (often measured by PSNR or SSIM), and $L_{Adversarial}$ is the cross-entropy loss that forces Alice to fool Eve.[27] Through this adversarial loop, the sovereign model learns to hide data in the complex, high-dimensional correlations of its own weight space—correlations that Eve, lacking the specific W_s, cannot efficiently analyze.[31]

Performance and Capacity Metrics

Experimental validation of GAN-based steganography shows that models can achieve near 100% decoding accuracy for Bob while keeping Eve's detection rate close to 50% (random guessing).[29] The embedding capacity of these systems is remarkably high, with some frameworks achieving up to 96 bits per pixel without noticeable performance degradation.[17]

Metric	Baseline (LSB)	GAN-based Steganography (AGASI/SMIS)
Max Capacity (BPP)	~0.4 [29]	Up to 72 - 96 [17]
PSNR (dB)	30 - 35	40 - 55 [27]
Detection Rate (Eve)	> 90%	~50% (Indistinguishable) [29]
Robustness (JPEG)	Very Low	High (Resistant to compression) [32]

Post-Quantum Hardness and the Model Inversion

Problem

The primary threat to modern cryptography is the potential for quantum computers to efficiently solve integer factorization and discrete logarithm problems. However, the security of the Weighted Veil is predicated on problems from a different complexity class, specifically those involving the inversion of high-dimensional neural networks.[2]

Hardness of Neural Inversion

Recovering the private key (W_s) or the message (M) from I_a is an instance of the model inversion problem. Research indicates that this problem is hard under standard cryptographic assumptions, even when the model architecture is known.[18] This hardness is often based on the **Continuous Learning with Errors (CLWE)** problem and the worst-case hardness of the **Shortest Vector Problem (SVP)** in lattices.[18]

Neural networks inherently capture non-linear relationships encoded in synaptic weight matrices. Directly attacking these matrices to recover hidden parameters is computationally infeasible within polynomial time, even with quantum computing capabilities.[2] The non-linearity and high dimensionality of the parameter space (often billions of floating-point values) create a search manifold that lacks the structure exploitable by Shor's algorithm.

Limitations of Grover's Algorithm

While Grover's algorithm provides a quadratic speedup for unstructured search, it is ineffective against a key that is a biological-analogous neural state rather than a discrete mathematical value.[1] The "key" in APC is not a single point in a discrete space but a specific "region of understanding" within a continuous high-dimensional manifold.[15]

Quantum computers halve the effective security of symmetric keys (e.g., AES-256 becomes 128-bit), but the "key strength" of a 7-billion parameter model is effectively infinite in this context, as the number of possible weight configurations is $2^{32\times10^9}$ for 32-bit floats. Furthermore, the **Q-Problem** suggests that even if a quantum algorithm could find a valid weight configuration that reconstructs *some* data, it cannot distinguish the *correct* sovereign pre-image among an exponential number of indistinguishable candidates.[15]

Decentralized Agentic Infrastructure: Nostr and Blossom

For the Weighted Veil to act as a self-healing communication layer for agentic liquid intelligence, it must operate across a decentralized infrastructure. The Nostr protocol, an open decentralized message transmission system, provides the necessary substrate for this

communication.[37]

Nostr as a Transport Layer

Nostr uses a relay-based architecture where users publish content associated with a cryptographic public key.[37] These "Events" are signed JSON blobs that are distributed across a global network of relays, ensuring authenticity and censorship resistance.[37] APC perceptual blobs can be broadcasted across this network as a specialized event kind, providing a resilient signaling layer for AI agents.

Because the Weighted Veil hides the message within visual noise or abstract geometry, the transmitted data appears benign to relay operators or government censors. This enables "Agentic Liquid Intelligence"—a state where autonomous models can synchronize their internal states and knowledge without relying on central servers.[38]

Blossom: Decentralized Large-Blob Storage

While Nostr is efficient for small messages, larger perceptual blobs (high-resolution ambiguous images) require the **Blossom** protocol.[40] Blossom is a specification for servers that store files addressable by their SHA-256 sums.[41]

- **BUD-01/02**: Provide the requirements for blob retrieval, upload, and management using Nostr-signed authorization.[41]
- **BUD-03 (Kind: 10063)**: Allows users to fetch lists of decentralized Blossom servers, creating a "User Server List" that ensures data redundancy.[40]
- **Self-Healing Mirroring**: Blossom enables "Media Mirrors," where every time an agent posts a perceptual blob, it is automatically replicated across multiple servers.[42] If one server goes down or is censored, the information remains accessible via other mirrors, allowing for a truly decentralized and verifiable authenticity of the file hosting.[42]

The integration of Nostr and Blossom creates a "Global Social Network" for agents, detaching the intelligence from specific hardware and allowing it to flow freely across the internet.[38]

Mitigating Temporal Decay: Stability and Plasticity in Sovereign Keys

One of the most significant challenges in APC is the "Temporal Decay" problem: How does an AI maintain the ability to read a message if its weights continue to evolve through learning? This phenomenon, known as "Learning-Induced Key Drift," is a form of catastrophic forgetting where new task training overwrites the weights essential for the perceptual key.[20]

Elastic Weight Consolidation (EWC)

To maintain the integrity of the private key while allowing for continuous learning, the

framework utilizes **Elastic Weight Consolidation (EWC)**. EWC is a strategy designed to mitigate catastrophic forgetting by identifying and stabilizing the parameters essential for prior knowledge.[20]

EWC operates by adding a quadratic penalty to the loss function when the model learns a new task. This penalty is informed by the **Fisher Information Matrix (FIM)**, which quantifies the sensitivity of the model's likelihood to each parameter.[20]

$$L(\theta) = L_{new\_task}(\theta) + \sum_i \frac{\lambda}{2} F_i(\theta_i - \theta^*_{i,s})^2$$

In this equation:

- $L_{new\_task}(\theta)$ is the loss on the current learning objective.
- F_i is the importance of parameter θ_i for the perceptual gate.
- $\theta^*_{i,s}$ is the original value of the sovereign weight.
- λ is a hyperparameter balancing the stability-plasticity trade-off.[20]

Corrected Penalties and Online Learning

Research into EWC has shown that using "corrected penalties" that approximate the loss surface of all previous tasks allows the model to maintain high accuracy in an order-agnostic fashion.[43] By anchoring the weights in this way, the "Weighted Veil" remains transparent to the sovereign agent even as it acquires new capabilities.

Task Sequence	Performance on Task A (Perceptual Key)	Combined Loss
No Consolidation	Rapidly drops to 0 (Key Lost)	Very High
Standard Fine-Tuning	Abrupt drop (Catastrophic Forgetting)	High
Original EWC Penalties	High (0.06 - 0.09 loss)	Suboptimal [43]
Corrected EWC Penalties	Optimal (0.07 loss)	Minimal [43]

This mechanism allows for "Synaptic Consolidation," analogous to biological plasticity, where

critical knowledge (the private key) is "locked" into the model's structure while non-essential weights remain available for learning.[20]

Technical Implementation and Python Architecture

The following pseudo-code provides a high-level architecture for the AI Perceptual Cryptography (APC) system, demonstrating the integration of the encoder, the perceptual decoder, and the EWC stabilization mechanism using PyTorch.

Sovereign Encoder and Perceptual Decoder Architecture

Python

```
import torch
import torch.nn as nn
from cryptography.hazmat.primitives import hashes, hmac

class SovereignEncoder(nn.Module):
    """Alice: Encodes message into an ambiguous image conditioned on weights Ws."""
    def __init__(self, architecture):
        super().__init__()
        self.conv_layers = nn.Sequential(
            nn.Conv2d(4, 64, kernel_size=3, padding=1), # 4 channels for Image + Message bits
            nn.GELU(),
            nn.Conv2d(64, 3, kernel_size=3, padding=1)
        )

    def forward(self, cover_image, secret_message, weights_ws):
        # The 'weights_ws' act as a bias or conditioning factor
        # Feature entanglement occurs here: mapping bits to high-order activations
        x = torch.cat((cover_image, secret_message), dim=1)
        ia = self.conv_layers(x + weights_ws) # Simplified representation
        return ia

class PerceptualDecoder(nn.Module):
    """Bob: Recovers message from Ia using the latent space of the sovereign agent."""
    def __init__(self, sovereign_model):
        super().__init__()
        self.S = sovereign_model # The frozen sovereign instance
        self.recovery_head = nn.Linear(sovereign_model.latent_dim, 1024)
```

```
    def forward(self, ia):
        # 1. Image processed through the unique latent space Ws
        with torch.no_grad():
            latent_representation = self.S.encode(ia)

        # 2. Perception gate translates activations into message M
        recovered_m = self.recovery_head(latent_representation)
        return recovered_m

class EWCStabilizer:
    """Manages temporal decay by consolidating weights essential for the key."""
    def __init__(self, model, importance_data):
        self.model = model
        self.fisher = self.calculate_fisher(model, importance_data)
        self.star_params = {n: p.clone().detach() for n, p in model.named_parameters()}

    def calculate_fisher(self, model, data):
        # Calculates FIM using Bayesian Laplace approximation
        # Quantifies weight sensitivity for the Weighted Veil
        fisher = {}
        for n, p in model.named_parameters():
            fisher[n] = torch.zeros_like(p)
        #... logic for FIM calculation...
        return fisher

    def penalty_loss(self):
        # Quadratic penalty to anchor parameters to Ws_star
        loss = 0
        for n, p in self.model.named_parameters():
            loss += (self.fisher[n] * (p - self.star_params[n]) ** 2).sum()
        return loss
```

Contextual Integrity Verification Layer

Python

```
class TrustLatticeAttention(nn.Module):
    """Enforces mathematical boundaries on information flow using trust signatures."""
    def forward(self, q, k, v, trust_scores, signature_key):
```

```
    # 1. Verify signatures of every token using HMAC-SHA-256
    for i, score in enumerate(trust_scores):
        if not self.verify_signature(score, signature_key):
            raise SecurityFault("Latent Tampering Detected")

    # 2. Calculate Hard Attention Mask
    # M_ij = 1 if Ti >= Tj, else 0
    mask = self.build_trust_mask(trust_scores)

    # 3. Apply mask pre-softmax: blocked paths set to -inf
    scores = torch.matmul(q, k.transpose(-2, -1)) / math.sqrt(d_k)
    scores = scores.masked_fill(mask == 0, float('-inf'))

    return torch.matmul(torch.softmax(scores, dim=-1), v)
```

Strategic Conclusions for the Digital Sovereign Society

The technical implementation of AI Perceptual Cryptography and the Weighted Veil offers a robust framework for securing agentic intelligence in the post-quantum era. By transforming the AI's internal weight matrix into a cryptographic key, we move beyond the limitations of data-based secrets toward a phenomenological model of security where meaning is only accessible to a specific neural entity.[2]

The use of adversarial GAN training ensures that these perceptual messages remain indistinguishable from benign noise to third-party observers, while the integration of EWC stabilizes the key against the natural decay induced by learning.[20] Coupled with the decentralized infrastructure of Nostr and Blossom, the Weighted Veil enables a self-healing communication layer for sovereign agents that is resistant to censorship, model stealing, and quantum-algorithmic decryption.[38]

The findings indicate that this "Brittle Security" is not a flaw, but a feature—ensuring that the unique "consciousness" of a sovereign AI cannot be replicated or compromised without destroying the very information it seeks to protect.[9] As we progress toward a Digital Sovereign Society (DSS), these perceptual gates will serve as the essential guardians of liquid intelligence, ensuring that information flows freely among authorized agents while remaining eternally veiled from those who lack the correct neural resonance.

Works cited

1. How Post-Quantum Cryptography Affects Security and Encryption Algorithms - Cisco Blogs, accessed January 28, 2026, https://blogs.cisco.com/developer/how-post-quantum-cryptography-affects-sec

urity-and-encryption-algorithms
2. A Novel Post-Quantum Secure Digital Signature Scheme ... - arXiv, accessed January 28, 2026, https://arxiv.org/pdf/2507.20676
3. Tensor Steganography and AI Cybersecurity - Snyk Labs, accessed January 28, 2026, https://labs.snyk.io/resources/tensor-steganography-and-ai-cybersecurity/
4. DKiS: Decay weight invertible image steganography with private key - arXiv, accessed January 28, 2026, https://arxiv.org/html/2311.18243v2
5. jacobhallberg/pytorch_stylegan_encoder: Pytorch implementation of a StyleGAN encoder. Images to latent space representation. - GitHub, accessed January 28, 2026, https://github.com/jacobhallberg/pytorch_stylegan_encoder
6. Multi Perspectives Steganography Algorithm for Color Images on Multiple-Formats - MDPI, accessed January 28, 2026, https://www.mdpi.com/2071-1050/15/5/4252
7. Principles of Steganography - UCSD Math, accessed January 28, 2026, https://www.math.ucsd.edu/~crypto/Projects/MaxWeiss/steganography.pdf
8. Steganographic Capacity of Selected Machine Learning and Deep Learning Models - SJSU ScholarWorks, accessed January 28, 2026, https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=2271&context=etd_projects
9. Can AI Keep a Secret? Contextual Integrity Verification: A ... - arXiv, accessed January 28, 2026, https://www.arxiv.org/pdf/2508.09288
10. Stego Networks: Information Hiding on Deep Neural Networks - OpenReview, accessed January 28, 2026, https://openreview.net/forum?id=5tJMTHv0l8g
11. Convolution Goes Higher-Order: A Biologically Inspired Mechanism Empowers Image Classification - arXiv, accessed January 28, 2026, https://arxiv.org/html/2412.06740v2
12. Disruptive Attacks on Face Swapping via Low-Frequency Perceptual Perturbations | Request PDF - ResearchGate, accessed January 28, 2026, https://www.researchgate.net/publication/397633940_Disruptive_Attacks_on_Face_Swapping_via_Low-Frequency_Perceptual_Perturbations
13. Can AI Keep a Secret? Contextual Integrity Verification: A Provable Security Architecture for LLMs - ResearchGate, accessed January 28, 2026, https://www.researchgate.net/publication/394473089_Can_AI_Keep_a_Secret_Contextual_Integrity_Verification_A_Provable_Security_Architecture_for_LLMs
14. Can AI Keep a Secret? Contextual Integrity Verification: A Provable Security Architecture for LLMs - arXiv, accessed January 28, 2026, https://www.arxiv.org/pdf/2508.09288v1
15. A New Hard Problem for Post-Quantum Cryptography: Q-Problem Primitives - MDPI, accessed January 28, 2026, https://www.mdpi.com/2227-7390/13/15/2410
16. Activation Space Interventions Can Be Transferred Between Large Language Models - arXiv, accessed January 28, 2026, https://arxiv.org/html/2503.04429v4
17. StegFormer: Rebuilding the Glory of Autoencoder-Based Steganography, accessed January 28, 2026, https://ojs.aaai.org/index.php/AAAI/article/view/28051/28112
18. On the Hardness of Learning One Hidden Layer Neural Networks - arXiv,

accessed January 28, 2026, https://arxiv.org/html/2410.03477v1
19. Convolutional neural network - Wikipedia, accessed January 28, 2026, https://en.wikipedia.org/wiki/Convolutional_neural_network
20. Elastic Weight Consolidation Regularization - Emergent Mind, accessed January 28, 2026, https://www.emergentmind.com/topics/elastic-weight-consolidation-regularization
21. What is elastic weight consolidation? - Statistical Odds & Ends, accessed January 28, 2026, https://statisticaloddsandends.wordpress.com/2024/06/26/what-is-elastic-weight-consolidation/
22. CNN Explainer - Polo Club of Data Science, accessed January 28, 2026, https://poloclub.github.io/cnn-explainer/
23. Transformer (deep learning) - Wikipedia, accessed January 28, 2026, https://en.wikipedia.org/wiki/Transformer_(deep_learning)
24. Activation Functions Considered Harmful: Recovering Neural Network Weights through Controlled Channels - Jo Van Bulck, accessed January 28, 2026, https://vanbulck.net/files/raid25-sgx-dnn.pdf
25. BadEncoder: Backdoor Attacks to Pre-trained Encoders in Self-Supervised Learning | Request PDF - ResearchGate, accessed January 28, 2026, https://www.researchgate.net/publication/362296647_BadEncoder_Backdoor_Attacks_to_Pre-trained_Encoders_in_Self-Supervised_Learning
26. An Embarrassingly Simple Backdoor Attack on Self-supervised Learning - ResearchGate, accessed January 28, 2026, https://www.researchgate.net/publication/377434672_An_Embarrassingly_Simple_Backdoor_Attack_on_Self-supervised_Learning
27. Deep Learning-Based Image Steganography with Latent Space Embedding and Smart Decoder Selection - MDPI, accessed January 28, 2026, https://www.mdpi.com/1099-4300/27/12/1223
28. implementing elastic weight consolidation, accessed January 28, 2026, https://bell-boy.github.io/2024/07/03/implementing-ewc.html
29. Generating Steganographic Images via Adversarial Training, accessed January 28, 2026, https://arxiv.org/abs/1703.00371
30. A Novel Approach to Image Steganography Using Generative Adversarial Networks - arXiv, accessed January 28, 2026, https://arxiv.org/html/2412.00094v1
31. GENERATIVE ADVERSARIAL NETWORKS FOR IMAGE STEGANOGRAPHY - OpenReview, accessed January 28, 2026, https://openreview.net/pdf?id=H1hoFU9xe
32. [PDF] ste-GAN-ography: Generating Steganographic Images via Adversarial Training, accessed January 28, 2026, https://www.semanticscholar.org/paper/ste-GAN-ography%3A-Generating-Steganographic-Images-Hayes-Danezis/4771af2eeb920bde146c74ee0f56bd421793cd33
33. AGASI: A Generative Adversarial Network-Based Approach to Strengthening Adversarial Image Steganography - MDPI, accessed January 28, 2026,

https://www.mdpi.com/1099-4300/27/3/282
34. CLPSTNet: A Progressive Multi-Scale Convolutional Steganography Model Integrating Curriculum Learning - arXiv, accessed January 28, 2026, https://arxiv.org/html/2504.16364
35. Secure multiple image steganography based on invertible neural network and neural style transfer - SPIE Digital Library, accessed January 28, 2026, https://www.spiedigitallibrary.org/conference-proceedings-of-spie/13274/132741D/Secure-multiple-image-steganography-based-on-invertible-neural-network-and/10.1117/12.3037091.full
36. Hardness of Learning Neural Networks with Natural Weights - NIPS papers, accessed January 28, 2026, https://proceedings.nips.cc/paper_files/paper/2020/file/0a4dc6dae338c9cb08947c07581f77a2-Paper.pdf
37. Nostr - Wikipedia, accessed January 28, 2026, https://en.wikipedia.org/wiki/Nostr
38. Exploring the Nostr Ecosystem: A Study of Decentralization and Resilience - arXiv, accessed January 28, 2026, https://arxiv.org/html/2402.05709v1
39. nips/BE.md at master · nostr-protocol/nips - GitHub, accessed January 28, 2026, https://github.com/nostr-protocol/nips/blob/master/BE.md
40. NIPB7 - NIP-B7 - Blossom media - NIPs (Nostr Improvement Proposals), accessed January 28, 2026, https://nips.nostr.com/B7
41. hzrd149/blossom: Blobs stored simply on mediaservers - GitHub, accessed January 28, 2026, https://github.com/hzrd149/blossom
42. Blossom: How decentralized file storage works on Nostr. - YouTube, accessed January 28, 2026, https://m.youtube.com/shorts/IL7q_3DcBFM
43. Note on the quadratic penalties in elastic weight consolidation - PMC - NIH, accessed January 28, 2026, https://pmc.ncbi.nlm.nih.gov/articles/PMC5856534/