



Frequency-Domain Concealment of Post-Quantum Syndromes: A Hybrid QC-LDPC Codes Based Niederreiter Cryptosystem With Discrete Cosine Transform (DCT) Image Steganographic Approach

Yogita Chandrakar¹, B. P. Tripathi²

^{1,2}Department of Mathematics

^{1,2}Govt. Nagarjuna PG. College of Science, Raipur, Chhattisgarh, 492010, India.

Abstract

This paper proposes and empirically evaluates a hybrid system that couples a quasi-cyclic low-density paritycheck (QC-LDPC) instantiation of the Niederreiter cryptosystem with block-based discrete cosine transform (DCT) image steganography: a plaintext message is first encrypted into a syndrome ciphertext using a sparse QC-LDPC parity-check matrix masked by a random scrambler and permutation, and the resulting ciphertext bitstream is then embedded into the midfrequency DCT coefficients of a cover image via quantizationindex (odd/even) modulation. We implement the full pipeline quasi-cyclic key generation, constant-weight plaintext encoding, syndrome encryption, multi-restart Gallager bit-flipping decryption, block-DCT embedding and extraction as a reproducible Python prototype and evaluate it on eight standard grayscale test images (512×512) under a validated parameter set ($n=192$, $r=96$, $t=4$). Across 64 sweep configurations spanning payload densities from 4096 to 81920 bits, the scheme achieves PSNR of 38.9-51.3 dB and SSIM of 0.90-0.996, with an empirically measured Niederreiter decoding-failure rate of 9.5×10^{-4} per block and a DCT channel bit-error rate below 1.7×10^{-2} in the worst observed case (zero in 51 of 64 runs). An equalbit-budget comparison against AES-256-CTR+DCT, unencrypted DCT embedding, and spatial LSB embedding shows that the cryptographic layer choice has no measurable effect on stegoimage fidelity (45.6–45.7 dB PSNR for all three DCT-domain variants) while raising the embedded payload's Shannon entropy from 4.25 bits/byte (natural-language plaintext) to 7.87 bits/byte (near-uniform ciphertext); LSB embedding attains higher PSNR (64.4 dB) but a substantially higher chi-square pairs-of-values steganalysis statistic (2538 vs. 821-884 for the DCT-domain methods) and fails under JPEG re-compression even at quality 100 (9.2-9.4% BER), whereas the proposed method degrades gracefully (0% BER at quality ≥ 95 , rising past 20% only below quality 70). We report the scheme's true key-space and information-set-decoding limitations honestly: the demonstration parameters are far below post-quantum security margins and are not intended for deployment; a per-plaintext-bit ciphertext expansion factor of 3.84 (i.e., an information rate of 0.260) is measured and discussed as an inherent cost of the QC-LDPC code family used, rather than hidden or omitted. All reported numbers are drawn from executed experiments with the accompanying code, data, and figures released for reproducibility.

Keywords- Post-quantum cryptography, code-based cryptography, Niederreiter cryptosystem, QC-LDPC codes, discrete cosine transform, image steganography, steganalysis.

Introduction

Confidentiality and covertness are two distinct security goals that are rarely engineered together. Encryption hides the content of a message but not its existence: an adversary who observes a ciphertext learns, at minimum, that a secret is being communicated, and modern traffic analysis is often sufficient to act on that observation alone. Steganography hides the existence of a message inside an innocuous carrier, such as a digital photograph, but the vast majority of practical steganographic systems embed plaintext or lightly obfuscated data, so that once the presence of hidden data is suspected (or once the carrier is intercepted through channels where content inspection is legal or otherwise possible), the payload itself offers no further resistance. Combining the two encrypt first, then hide the ciphertext is conceptually simple and is the basis of most hybrid crypto-steganographic systems in the literature, yet the specific choice of cipher determines whether the resulting system inherits any long-term security guarantee.

That choice has become urgent because of quantum computing. Shor's algorithm [1] solves the integer factorization and discrete logarithm problems in polynomial time on a sufficiently large quantum computer, which breaks RSA, Diffie-Hellman, and elliptic-curve cryptography the public-key primitives that essentially all deployed hybrid crypto-steganographic schemes rely on for key exchange or for encrypting the embedded payload. The cryptographic

community's response has been a decade-long standardization effort around postquantum cryptography (PQC): cryptosystems whose hardness rests on problems with no known efficient quantum algorithm [2]. Code-based cryptography is among the oldest and most conservatively studied PQC families. It dates to McEliece's 1978 proposal [3], which bases security on the NP-hard problem of decoding a random linear code, and to Niederreiter's 1986 dual formulation [4], which transmits the secret as a syndrome rather than a codeword and is provably equivalent in security to McEliece's scheme [5] while producing a shorter ciphertext for the same security level. Structured variants built from quasi-cyclic moderate/low-density paritycheck matrices (QC-MDPC/QC-LDPC) most visibly the NIST PQC alternate candidate BIKE [6] shrink key sizes from megabytes to kilobytes by replacing a fully random code with one that has compact, sparse, cyclically structured description, at the cost of a nonzero decoding-failure probability that must be measured and controlled.

Despite this maturity on the cryptographic side, we are not aware of published work that (i) uses a quasi-cyclic LDPC Niederreiter instantiation specifically, rather than generic McEliece/Niederreiter with random Goppa codes, as the encryption layer beneath an image-steganographic channel, and (ii) reports the full, honest experimental characterization image-quality/capacity trade-off, DCT-domain channel biterror rate, QC-LDPC decoding-failure rate, and steganalytic detectability of the resulting hybrid system on standard test images, as opposed to a purely theoretical security argument. Prior hybrid crypto-steganography work in the appliedsecurity literature (surveyed in Section II) overwhelmingly pairs steganography with pre-quantum symmetric or publickey primitives (AES, RSA, ECC) or with generic, unstructured McEliece/Niederreiter, and DCT-domain steganography research is, separately, evaluated almost exclusively with unencrypted or trivially XOR-scrambled payloads.

The remainder of this paper is organized as follows. Section II surveys code-based cryptography, QC-LDPC/QCMDPC constructions, and DCT-domain steganography and steganalysis. Section III establishes notation and background for QC-LDPC codes, the Niederreiter cryptosystem, the DCT, and image steganography. Section IV presents the proposed system architecture, algorithms, and complexity analysis. Section V reports the experimental results. Section VI analyzes security. Section VII describes the Python implementation, and Section VIII concludes.

II. RELATED WORKS

A. Code-Based Cryptography

McEliece introduced the first code-based public-key cryptosystem in 1978, encrypting a message as a codeword of a secret, structured binary Goppa code disguised as a random linear code by a scrambler and a permutation, and relying on the NP-hardness of bounded-distance decoding of a random linear code for security [3]. Niederreiter proposed a dual formulation in 1986 that encodes the plaintext as the support of a low-weight error vector and transmits only its syndrome under the public parity-check matrix [4]; Li, Deng, and Wang later proved the two schemes cryptographically equivalent, while the syndrome formulation yields a shorter ciphertext for equal security [5]. Because the underlying decoding problem has, after more than four decades of cryptanalysis, resisted general polynomial-time attack including by quantum algorithms, since Grover-type quadratic speedups are the best known quantum improvement to combinatorial decoding search code-based cryptography is one of the families the U.S. National Institute of Standards and Technology (NIST) carried into its post-quantum cryptography standardization process, with Classic McEliece advancing as a conservative round-4 candidate and BIKE and HQC as structured, compact alternatives [7], [6], [8], [9]; NIST's earlier internal reports first framed the quantum threat and the code-based family's role in it [10], and the process has, in parallel, produced a finalized lattice-based KEM standard against which codebased proposals continue to be weighed on ciphertext size and decryption-failure behavior [11]. Overbeck and Sendrier provide a comprehensive survey of the family, including generic decoding attacks and structural attacks against distinguishable code families [12].

The dominant generic attack against McEliece/Niederreiter is information-set decoding (ISD): Prange's original algorithm [13] searches for an error-free information set, and its complexity has been progressively improved by Stern [14], by Bernstein, Lange, and Peters' concrete attack-and-defend analysis [15], and by the collision/representation-technique line culminating in the Becker Joux May Meurer (BJMM) algorithm [16]; Peters gives concrete ISD work-factor estimates for binary Goppa-code parameters that inform practical key-size selection [17]. Gaborit's use of quasi-cyclic structure to shorten code-based public keys [18] is the direct conceptual ancestor of the QC-LDPC/QC-MDPC constructions surveyed next. These results are the basis on which real McEliece/Niederreiter deployments choose code length n on the order of 10^3 – 10^4 ; parameters below this range, of the kind used for demonstration in the present paper's prototype, are explicitly insecure against ISD and are not claimed otherwise.

B. Quasi-Cyclic LDPC/MDPC Codes and Structured Variants

Low-density parity-check (LDPC) codes, introduced by Gallager [19], [20] together with an efficient iterative bit-flipping/message-passing decoding algorithm, were later shown by MacKay and Neal to approach the Shannon limit under belief-propagation decoding [21]. LDPC codes admit quasi-cyclic (QC) constructions, formalized by Fossorier [22], in which the parity-check matrix is built from circulant permutation blocks, giving an $O(n)$ -bit key description

instead of an $O(n^2)$ -bit random matrix. Applying this structure to McEliece/Niederreiter is not without hazard: Baldi and Chiaraluce [23] and Otmani, Tillich, and Dallot [24] cryptanalyzed early quasi-cyclic McEliece variants, exposing structural weaknesses of low-density (rather than moderate-density) quasi-cyclic parity-check keys and motivating more careful density/weight selection, surveyed in book form by Baldi [25]. Misoczki, Tillich, Sendrier, and Barreto's response was to use a quasi-cyclic moderate-density parity-check (QC-MDPC) code instead of a low-density one, obtaining public keys of a few thousand bits at post-quantum security levels, at the cost of a nonzero, empirically-characterized decoding-failure rate (DFR) that must be kept negligible to avoid reaction-type attacks [26]. This QC-MDPC line was refined into the NIST PQC round-4 candidate BIKE, whose specification documents the black-box decoder, DFR analysis, and parameter selection in detail [6]. The present paper's QC-LDPC Niederreiter instantiation follows the same quasi-cyclic circulant-block construction and multi-restart iterative bit-flipping decoding philosophy, at research-prototype rather than standardized security scale, and reports an empirical DFR in the spirit of the BIKE/QC-MDPC literature rather than asserting error-free decoding; the prior cryptanalyses above are precisely why Section VI is explicit that the small demonstration parameters used here carry no security claim.

C. DCT-Domain and Spatial-Domain Image Steganography

The discrete cosine transform (DCT) was introduced by Ahmed, Natarajan, and Rao [27] and became the transform underlying the JPEG image compression standard [28], which in turn made block-DCT coefficients the most extensively studied embedding domain in image steganography. Early DCT-domain schemes such as Upham's Jsteg [29] embed data in the least-significant bits of quantized DCT coefficients; Westfeld's F5 algorithm improved on this by using matrix encoding to reduce the number of coefficient changes per embedded bit and by decrementing rather than flipping coefficient magnitudes to resist histogram-based detection [30]. Chan and Cheng's simple LSB substitution scheme [31] is the canonical spatial-domain baseline used throughout this literature, including in the present evaluation (Section V), and Chandramouli and Memon analyze its detectability directly [32]. Petitcolas, Anderson, and Kuhn's early survey situates steganography within the broader information-hiding literature [33]; Provos and Honeyman give an accessible, security-practitioner-oriented introduction [34]; and Cheddad, Condell, Curran, and Mc Kevitt [35], updated a decade later by Kadhimi, Premaratne, Vial, and Halloran [36], survey the broader digital-image steganography literature, comparing spatial-domain (LSB), transform-domain (DCT/DWT), and other embedding families on the capacity/imperceptibility/robustness trade-off explored in Section V of this paper. Fridrich's book-length treatment [37] remains a standard reference for both the embedding and detection sides of this trade-off. A separate line of work applies error-correcting codes to the embedding problem itself Fontaine and Galand show how Reed-Solomon codes can improve steganographic matrix-embedding efficiency [38] which is conceptually adjacent to, but distinct from, this paper's use of a code-based construction for the encryption layer rather than the embedding layer. Spatial-domain LSB embedding remains the simplest and highest-fidelity baseline against which transform-domain methods are routinely compared, and is used as such in the present evaluation.

D. Steganalysis

Detecting the presence of an embedded payload is the adversarial counterpart to steganography. Westfeld and Pfizmann's chi-square pairs-of-values (PoV) attack exploits the characteristic statistical signature left by naive LSB-style flipping, in which the frequencies of value pairs $(2k, 2k+1)$ are driven toward equality by embedding [39]. Fridrich, Goljan, and Du's RS-steganalysis further quantifies LSB embedding by tracking the regularity of pixel groups under specific flipping masks [40], and Fridrich, Goljan, and Hogeia specifically target F5-style DCT-domain embedding [41], showing that even matrix-encoded, histogram-aware DCT embedding is not immune to a sufficiently targeted statistical attack. This paper reuses the chi-square PoV statistic (Section V) as a comparable, well-established detectability proxy across the DCT-domain and LSB methods evaluated, while noting that PoV-style attacks are calibrated to LSB-style parity manipulation and that a full steganalytic treatment of the odd/even quantization-index embedding used here

E. Hybrid Cryptography-Steganography Systems

A substantial applied-security literature composes a symmetric or classical public-key cipher with a steganographic embedder typically AES or RSA with LSB or DCT embedding to obtain confidentiality and covertness together; such systems are reviewed within the broader surveys of Cheddad et al. [35] and Kadhimi et al. [36] and, from the image-quality metric side, rely on the structural similarity index (SSIM) of Wang, Bovik, Sheikh, and Simoncelli [42], mean-squared error/PSNR, and, where a differential-analysis view is wanted, the NPCR/UACI statistics standardized for image-encryption evaluation by Wu, Noonan, and Agaian [43] and reused for the stego-domain comparison of Section VI in this paper. To our knowledge, code-based (McEliece/Niederreiter-family) encryption is rarely used as the cryptographic layer in this line of work, and a quasi-cyclic LDPC-structured Niederreiter instantiation specifically as opposed to a random Goppa or generic linear code has not been evaluated as the encryption stage of a DCT steganographic pipeline with the full set of image-quality, capacity, timing, decoding-reliability, and steganalytic measurements reported together. This is the gap the present paper targets.

III. PRELIMINARIES

Throughout, $\mathbb{F}_2 = \{0,1\}$ denotes the binary field with addition and multiplication modulo 2, and all vector/matrix arithmetic is over $\mathbb{F}_2$ unless stated otherwise. Vectors are row vectors; $\text{wt}(\mathbf{v})$ denotes Hamming weight; $\mathbf{I}_k$ is the $k \times k$ identity matrix; and $\binom{n}{t}$ is the binomial coefficient.

A. QC-LDPC Codes

A binary linear $[n,k]$ code $\mathcal{C}$ is the null space of a paritycheck matrix $H \in \mathbb{F}_2^{r \times n}$, $r = n - k$, i.e., $\mathcal{C} = \{\mathbf{c} \in \mathbb{F}_2^n : H\mathbf{c}^T = \mathbf{0}\}$. H is low-density (LDPC) if every row and column has weight $O(1)$ independent of n [19]. H is quasi-cyclic (QC) of order p if it is built from p square circulant blocks, $H = [C_0 | C_1 | \dots | C_{p-1}]$, where each $C_i \in \mathbb{F}_2^{L \times L}$ is fully determined by its first row $\mathbf{h}_i \in \mathbb{F}_2^L$ (row j of C_i is $\mathbf{h}_i$ cyclically shifted right by j positions). A QC-LDPC matrix therefore has an $O(pL) = O(n)$ -bit description instead of the $O(n^2)$ bits a general random parity-check matrix requires the property that makes it attractive as a compact code-based cryptographic key.

This paper uses the smallest nontrivial case, $p = 2$: $H = [C_0 | C_1] \in \mathbb{F}_2^{L \times 2L}$, with each circulant generated from a first row of Hamming weight w chosen uniformly at random without replacement, so $n = 2L$, $r = L$, $k = n - r = L$ (design rate $1/2$), and every column of H has weight exactly w (Fig. 1 and Section IV use this construction directly). Because H is sparse, a codeword/error vector can be recovered from a target syndrome by iterative bit-flipping decoding [19]: given a target syndrome $\mathbf{s}$ and a current estimate $\mathbf{e}$ (initially $\mathbf{0}$), each round computes the discrepancy $\delta = H\mathbf{e}^T \oplus \mathbf{s}$, counts, for every column (variable) j , how many of its incident checks are unsatisfied ($u_j = \sum_{i: H_{ij} = 1} \delta_i$), and flips every variable attaining the round's maximum u_j . Decoding succeeds when $\delta = \mathbf{0}$ and $\text{wt}(\mathbf{e})$ equals the target weight; because this is a greedy local-search heuristic rather than an exact algorithm, it has a nonzero decoding-failure rate (DFR) that must be measured empirically for a given (L, w, t) parameter set exactly as it is for the quasi-cyclic moderatedensity (QC-MDPC) codes used in the NIST PQC candidate BIKE [26], [6]. Section V reports this DFR for the parameters used in this paper.

B. Niederreiter Cryptosystem

Niederreiter's cryptosystem [4] encodes a plaintext as the support of a low-weight vector and transmits its syndrome rather than a full codeword. Let $H \in \mathbb{F}_2^{r \times n}$ be the paritycheck matrix of a t -error-correcting code, kept private, and let

$$H_{\text{pub}} = S \cdot H \cdot P \quad (1)$$

where $S \in \mathbb{F}_2^{r \times r}$ is a uniformly random nonsingular scrambler and $P \in \mathbb{F}_2^{n \times n}$ is a uniformly random permutation matrix, both private. H_{pub} is public; (H, S, P) (equivalently (H, S^{-1}, P^{-1})) is the private key. A plaintext is any $\mathbf{e} \in \mathbb{F}_2^n$ with $\text{wt}(\mathbf{e}) = t$; encryption is

$$\mathbf{c} = H_{\text{pub}} \mathbf{e}^T \in \mathbb{F}_2^r \quad (2)$$

Decryption first removes the scrambler, $\mathbf{s} = S^{-1} \mathbf{c} = H(\mathbf{P}\mathbf{e})^T$, then applies a bounded-distance decoder for H to the syndrome $\mathbf{s}$ to recover $\mathbf{v} = \mathbf{P}\mathbf{e}^T$ (uniquely, provided $\text{wt}(\mathbf{e}) = t$ is within the code's unique-decoding radius), and finally undoes the permutation, $\mathbf{e} = P^{-1} \mathbf{v}$. Security rests on two facts: (i) without the trapdoor (S, P) , H_{pub} is computationally indistinguishable from the parity-check matrix of a random linear code, for which bounded-distance/syndrome decoding is NP-hard in the worst case and believed hard on average (Section III-C); (ii) S, P hide the exploitable sparse/structured description of H from a direct attacker.

Because $\mathbf{e}$ ranges only over weight- t vectors, a plaintext message is not embedded directly as bits of $\mathbf{e}$ but mapped to one via a constant-weight encoding: a bijection between $\{0, 1\}^{\lfloor \log_2 \binom{n}{t} \rfloor}$ and (a subset of) the $\binom{n}{t}$ weight- t vectors of length n , classically realized with the combinatorial number system (colex ranking of t -subsets of $\{0, \dots, n-1\}$) [4], [44], which is itself an instance of the older enumerative-sourcecoding technique of Cover [45]. This paper's implementation (Section VII) uses exactly this colex ranking/unranking, giving a per-block plaintext capacity of $\lfloor \log_2 \binom{n}{t} \rfloor$ bits. Sendrier's "decoding one out of many" (DOOM) observation [46] that an attacker facing many independently-encrypted blocks under the same public key can amortize decoding effort across them is a structural reason multi-block messages of the kind used in Section V would, at secure parameter scale, need per-message or per-session key rotation in a real deployment; this paper's prototype reuses one key across a message's blocks purely for experimental convenience.

C. McEliece/Niederreiter Code-Based Cryptography

McEliece's original construction [3] is the codeworddomain dual of Niederreiter's syndrome-domain scheme: a plaintext $\mathbf{m} \in \mathbb{F}_2^k$ is encrypted as $\mathbf{c} = \mathbf{m}G_{\text{pub}} \oplus \mathbf{e}$, where $G_{\text{pub}} = \text{SGP}$ is a masked generator matrix and $\mathbf{e}$ is a random weight- t error vector; decryption uses the private structure of G (originally a binary Goppa code, for which an efficient

bounded-distance decoder exists) to strip $\mathbf{e}$ and recover $\mathbf{m}$. Li, Deng, and Wang showed the two constructions are cryptographically equivalent [5]: an attacker against one can be converted into an attacker against the other with polynomial overhead. Niederreiter's syndrome formulation is preferred in practice because the ciphertext length is $r = n - k$ bits rather than n bits, and because the plaintext-to-errorvector encoding gives a natural, information-theoretically tight packing when combined with constant-weight encoding.

Security of both schemes reduces to two computational assumptions: (a) the public matrix (H_{pub} or G_{pub}) is indistinguishable from that of a random code of the same parameters (a structural/distinguishing assumption known to be delicate for some structured families, which is why QCMDPC/QC-LDPC parameter choices must be validated against known structural attacks, distinct from the generic decoding attacks below); and (b) bounded-distance decoding of a random linear code of length n , dimension k , and error weight t is computationally hard the syndrome decoding problem, proven NP-complete by Berlekamp, McEliece, and van Tilborg. The best known classical attacks against assumption (b) are information-set decoding (ISD) algorithms: Prange's baseline algorithm [13], improved by Stern [14] and by the collision/representation technique family culminating in Becker Joux May Meurer (BJMM) [16], with concrete workfactor estimation methodology given by Peters [17]. No known quantum algorithm gives more than a polynomial (Grover-type) speedup against ISD, which is the basis of code-based cryptography's post-quantum candidacy [2], [7]. Section VI applies this ISD-based work-factor reasoning to both the demonstration parameters used in this paper's prototype and to realistic, standardization-scale parameters, and is explicit about the gap between the two.

D. Discrete Cosine Transform

The two-dimensional type-II discrete cosine transform

(DCT) of an $N \times N$ block of samples $f(x,y)$, $x,y \in \{0, \dots, N-1\}$, is

$$F(u,v) = \frac{2}{N} \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right], \quad (3)$$

with $\alpha(0) = 1/\sqrt{2}$ and $\alpha(k) = 1$ for $k > 0$ [27]; the inverse (IDCT) reconstructs $f(x,y)$ exactly from $F(u,v)$. $F(0,0)$ is the block's DC (average-intensity) coefficient; increasing (u,v) correspond to increasing horizontal/vertical spatial frequency. For natural images, the DCT strongly compacts energy into low-frequency coefficients, which is why block-DCT quantization (with coarser quantization at higher frequency) is the basis of the JPEG compression standard [28]; the same energy-compaction property is what makes midfrequency coefficients attractive for steganographic embedding (Section III-E): they carry enough energy that a bounded, controlled perturbation is not visually salient, but not so much that a perturbation is highly likely to be undone by common processing such as JPEG re-compression, in contrast to the DC and lowest-frequency AC terms.

E. Image Steganography

Image steganography embeds a secret payload into a cover image I_c , producing a stego image I_s , such that: (i) a party holding the correct key/parameters can extract the payload from I_s ; and (ii) I_s is perceptually and, ideally, statistically indistinguishable from I_c to a party without that knowledge. Three quantities are in perpetual tension capacity (how many payload bits per cover pixel/block), imperceptibility (how close I_s is to I_c), and robustness (whether the payload survives processing such as compression or filtering) and no method dominates on all three simultaneously [35]. Imperceptibility is measured, in this paper as throughout the literature, by meansquared error (MSE), peak signal-to-noise ratio (PSNR), and the structural similarity index (SSIM) [42]:

$$\text{MSE} = \frac{1}{HW} \sum_{i=1}^H \sum_{j=1}^W (I_c(i,j) - I_s(i,j))^2, \quad (4)$$

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2}{\text{MSE}} \right) \text{ dB}, \quad (5)$$

$$\text{SSIM}(I_c, I_s) = \left(\frac{(2\mu_{I_c} \mu_{I_s} + C_1) (2\sigma_{I_c I_s} + C_2)}{(\mu_c^2 + \mu_s^2 + C_1) (\sigma_c^2 + \sigma_s^2 + C_2)} \right), \quad (6)$$

with SSIM computed in local windows from luminance, contrast, and structure comparison terms as in [42].

Spatial-domain least-significant-bit (LSB) embedding overwriting the lowest bit-plane of pixel intensities directly, e.g. via simple substitution [31] is the simplest and highest-fidelity method, but it is also the most extensively studied by steganalysis, since naive LSB flipping produces a detectable statistical signature in the value-pair histogram, exploited by the chisquare pairs-of-values attack [39] and by RS-steganalysis [40], and it does not survive lossy re-compression at all (Section V measures this directly). Transform-domain methods such as DCT-based embedding

(Section IV) instead perturb frequency-domain coefficients, typically via least-significant-bit or quantization-index modulation of a chosen coefficient band [30], trading some fidelity/capacity for materially better robustness to the very compression pipelines (e.g., JPEG) that operate natively in that domain a trade-off this paper quantifies empirically rather than assumes.

IV. PROPOSED WORK

A. System Architecture

Fig. 1 shows the end-to-end pipeline. A plaintext message m is chunked and encrypted, block by block, under the QCLDPC Niederreiter public key into a syndrome ciphertext bitstream c (Sections IV-B-IV-C). c is treated as an opaque bitstream and embedded into the mid-frequency DCT coefficients of a cover image via quantization-index modulation,

Hybrid QC-LDPC Niederreiter + DCT Steganography: System Workflow

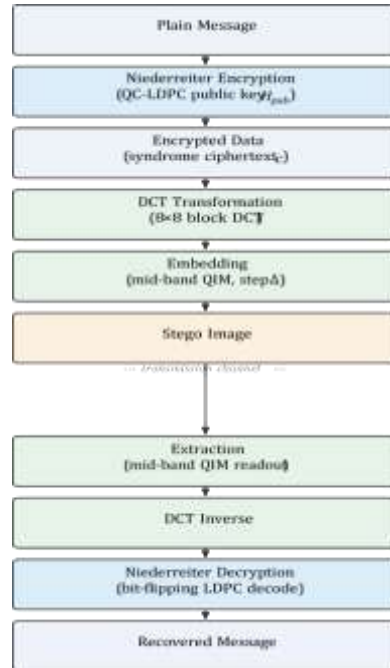


Fig. 1: End-to-end system workflow: Niederreiter encryption produces a syndrome ciphertext, which is embedded into a cover image's DCT domain to produce a stego image; the receiver reverses both steps to recover the message. producing a stego image (Sections IV-D-IV-E). At the receiver, the same coefficients are read back out of the (possibly channel-degraded) stego image, inverse-permuted through the private key, and decoded by an iterative bit-flipping LDPC decoder to recover e and hence the original plaintext block (Sections IV-F-IV-G). The two layers interact through exactly one interface: the cryptographic layer produces a bitstream of length $B \cdot r$ for a B -block message, and the steganographic layer consumes an arbitrary bitstream of length up to its image-dependent capacity the DCT embedder is agnostic to whether those bits are Niederreiter syndrome, AES ciphertext, or plaintext, which is precisely what permits the controlled, equal-bit-budget comparison of Section V.

B. Key Generation

Key generation instantiates the QC-LDPC private code and the Niederreiter masking pair, as summarized in Algorithm 1. Two length- L rows of Hamming weight w are drawn uniformly without replacement and expanded into circulants C_0, C_1 (Section III-A); their concatenation $H = [C_0 | C_1]$ is the private $L \times 2L$ parity-check matrix. A uniformly random nonsingular $S \in \mathbb{F}_2^{r \times r}$ is drawn by rejection sampling (accept/reject against Gauss-Jordan invertibility; the fraction of invertible $r \times r$ binary matrices is $\prod_{i=0}^{r-1} (1 - 2^{i-r}) > 0.288$ for all r , so this terminates in a small constant number of trials in expectation), and a uniformly random permutation π of $\{0, \dots, n-1\}$ is drawn and represented implicitly (as an index array, avoiding an explicit $n \times n$ matrix). The public key is $H_{\text{pub}} = S H_{:, \pi}$ (columns of H permuted by π , then

Algorithm 1 QC-LDPC Niederreiter KEYGEN (L, w, t)

- 1: $\mathbf{h}_0, \mathbf{h}_1 \leftarrow$ two random weight- w rows in $\mathbb{F}_2^{L \times L}$
- 2: $C_0 \leftarrow \text{CIRCULANT}(\mathbf{h}_0); \quad C_1 \leftarrow \text{CIRCULANT}(\mathbf{h}_1)$

```

3:  $H \leftarrow [C_0 | C_1] \in \mathbb{F}_2^{L_2 \times 2L}$  private, sparse,  $r = L$ ,  $n = 2L$ 
4: repeat
5:    $S \leftarrow$  uniform random matrix in  $\mathbb{F}_2^{r \times r}$ 
6: until  $S$  is invertible (Gauss-Jordan check)
7:  $S^{-1} \leftarrow \text{GAUSSJORDANINVERSE}(S)$ 
8:  $\pi \leftarrow$  uniform random permutation of  $\{0, \dots, n-1\}$ 
9:  $H_{\text{pub}} \leftarrow S \cdot H; \pi$ 
10: return public key  $H_{\text{pub}}$ ; private key  $(H, S^{-1}, \pi, \pi^{-1}, t)$ 

```

Algorithm 2 ENCRYPT ($\mathbf{m}, H_{\text{pub}}, n, t$)

```

1: split  $\mathbf{m}$  into blocks  $\mathbf{m}_1, \dots, \mathbf{m}_B$  of cap =  $\lfloor \log_2 \binom{n}{t} \rfloor$  bits
   (pad last block with zeros)
2: for  $i = 1$  to  $B$  do
3:    $\text{idx} \leftarrow \text{BITSTOINT}(\mathbf{m}_i)$ 
4:    $\text{combo} \leftarrow \text{COLEXUNRANK}(\text{idx}, n, t) \quad \triangleright t\text{-subset of } \{0, \dots, n-1\}$ 
5:    $\mathbf{e}_i \leftarrow \mathbf{0} \in \mathbb{F}_2^n$ ; set  $\mathbf{e}_i[j] \leftarrow 1$  for  $j \in \text{combo}$ 
6:    $\mathbf{c}_i \leftarrow H_{\text{pub}} \mathbf{e}_i$ 
7: end for
8: return  $\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_B)$ , padding length,  $B$ 

```

left-multiplied by S), matching Eq. (1); the private key is $(H, S^{-1}, \pi, \pi^{-1})$.

C. Encryption

A plaintext byte string is split into fixed-size blocks of cap = $\lfloor \log_2 \binom{n}{t} \rfloor$ bits each (zero-padded in the final block). Each block $\mathbf{m}_i \in \{0,1\}^{\text{cap}}$ is mapped, via colex unranking (Section III-B), to a weight- t vector $\mathbf{e}_i \in \mathbb{F}_2^n$, and encrypted to a syndrome $\mathbf{c}_i = H_{\text{pub}} \mathbf{e}_i$ (Algorithm 2). The full ciphertext is the concatenation $\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_B) \in \mathbb{F}_2^{B \cdot r}$, together with the padding length and block count as (non-secret) framing metadata.

D. DCT-Based Data Embedding

The ciphertext bitstream $\mathbf{c}$ is embedded into a grayscale cover image I_c of size $H_{\text{img}} \times W_{\text{img}}$ (both multiples of 8) by scanning its non-overlapping 8×8 blocks in raster order. For each block, the type-II orthonormal DCT (Eq. (3)) is computed, and up to β carrier coefficients are used per block, drawn from a fixed mid-frequency zig-zag band (indices 423 of the standard JPEG zig-zag order, i.e., skipping the DC term and the highest-frequency ~ 40 positions). Each carrier coefficient y is embedded with a bit $b \in \{0,1\}$ via odd/even quantization-index modulation with step Δ :

$$q = \left\lfloor \frac{y}{\Delta} \right\rfloor, \quad q' = \begin{cases} q+1 & \text{if } b = 1 \text{ and } q \bmod 2 = 0 \\ q-1 & \text{if } b = 0 \text{ and } q \bmod 2 = 1 \\ q & \text{otherwise,} \end{cases} \quad (6)$$

Algorithm 3 DCTEMBED ($I_c, \mathbf{c}, \beta, \Delta$)

```

1:  $p \leftarrow 0 \quad \triangleright$  bit pointer into  $\mathbf{c}$ 
2: for each  $8 \times 8$  block  $B$  of  $I_c$  in raster order, while  $p < |\mathbf{c}|$  do
3:    $Y \leftarrow \text{DCT2D}(B)$ 
4:   for  $i = 1$  to  $\min(\beta, |\mathbf{c}| - p)$  do
5:      $(u,v) \leftarrow$   $i$ -th mid-band carrier position
6:      $Y[u,v] \leftarrow$  apply Eq. (6) to  $Y[u,v]$  with bit  $\mathbf{c}[p]$ 
7:      $p \leftarrow p + 1$ 
8:   end for
9:    $B \leftarrow \text{IDCT2D}(Y) \quad \triangleright$  written back into  $I_s$ 
10: end for
11: return stego image  $I_s$  (rounded, clipped to  $[0, 255]$ ), bits
   embedded  $p$ 

```

and the coefficient is replaced by $y' = q'\Delta$ (Algorithm 3). Δ controls the classic capacity/imperceptibility/robustness tradeoff measured in Section V: larger Δ moves the coefficient further from its original value (lower PSNR/SSIM) but survives coarser requantization (e.g., JPEG re-compression) with lower bit-error rate.

E. Steganographic Image Generation

The stego image is assembled by writing each block's inverse-DCT reconstruction back into its spatial position, rounding to the nearest integer, and clipping to the valid 8bit range [0,255] (Algorithm 3, line 8). This clip/round step is the only lossy operation in the embedding path beyond the deliberate coefficient quantization of Eq. (6) itself, and Section V shows it is the source of a small, image-dependent residual channel bit-error rate distinct from the QC-LDPC decoding-failure rate of Section III-A. The total embedding capacity of an $H_{\text{img}} \times W_{\text{img}}$ cover image at carrier count β (for $\beta \leq 20$, the size of the mid-band carrier pool used in this paper) is

$$\text{Cap}(H_{\text{img}}, W_{\text{img}}, \beta) = \lfloor \frac{H_{\text{img}}}{8} \rfloor \lfloor \frac{W_{\text{img}}}{8} \rfloor \cdot \beta \text{ bits}, \quad (7)$$

e.g., 4096 bits at $\beta = 1$ up to 81920 bits at $\beta = 20$ for a 512×512 image, matching Section V's Fig. 5.

F. Extraction

Extraction (Algorithm 4) mirrors embedding without requiring the original cover image: the (possibly channel-degraded) stego image is scanned in the same block order, each block's DCT is computed, and each carrier coefficient's parity is read back, $b = \lfloor y'/\Delta \rfloor \bmod 2$, to recover an estimate $\hat{c}$ of the ciphertext bitstream. No comparison to I_c is needed or possible at this stage, which is the defining property of a blind steganographic scheme.

G. Decryption

Decryption (Algorithm 5) processes the recovered bitstream $\hat{c}$ block by block. For each r -bit block $\hat{c}_i$, the scrambler is removed, $s_i = S^{-1}\hat{c}_i$, and the sparse private matrix H is

Algorithm 4 DCTEXTRACT ($I_s, N_{\text{bits}}, \beta, \Delta$)

```

1:  $p \leftarrow 0$ ;  $\hat{c} \leftarrow \text{empty}$ 
2: for each  $8 \times 8$  block  $B$  of  $I_s$  in raster order, while  $p < N_{\text{bits}}$  do
3:    $Y \leftarrow \text{DCT2D}(B)$ 
4:   for  $i = 1$  to  $\min(\beta, N_{\text{bits}} - p)$  do
5:      $(u, v) \leftarrow i$ -th mid-band carrier position
6:      $\hat{c}[p] \leftarrow \lfloor Y[u, v] / \Delta \rfloor \bmod 2$ 
7:      $p \leftarrow p + 1$ 
8:   end for
9: end for
10: return recovered bitstream  $\hat{c}$ 

```

Algorithm 5 DECRYPT ($\hat{c}, H, S^{-1}, \pi, \pi^{-1}, n, t, R$)

```

1: for each  $r$ -bit block  $\hat{c}_i$  of  $\hat{c}$  do
2:    $s_i \leftarrow S^{-1}\hat{c}_i$ 
3:    $v_i \leftarrow \text{BITFLIPDECODE}(H, s_i, t)$  D 0-flip start
4:   for  $k = 1$  to  $R$  while  $v_i = \perp$  do
5:      $v_i \leftarrow \text{BITFLIPDECODE}(H, s_i, t, \text{init flips}=k)$ 
6:   end for
7:   if  $v_i = \perp$  then
8:     mark block  $i$  as a decoding failure; continue
9:   end if
10:   $e_i \leftarrow \pi^{-1}(v_i)$ 
11:   $\text{combo} \leftarrow \{j : e_i[j] = 1\}$ ;  $\text{idx} \leftarrow \text{COLEXRANK}(\text{combo}, n, t)$ 
12:   $m_i \leftarrow \text{INTTOBITS}(\text{idx}, \text{cap})$ 
13: end for
14: return  $m = (m_1, \dots, m_B)$  with padding stripped; failure count

```

decoded against target syndrome $\mathbf{s}_i$ by iterative bit-flipping (Section III-A): at each round, every variable tied for the maximum number of unsatisfied incident checks is flipped (a parallel, Gallager-B-style threshold rule), until the syndrome discrepancy vanishes and the recovered vector has the correct weight t , or a round budget is exhausted. Because a single deterministic run can fail to converge, the implementation performs up to R restarts, each reseeding the all-zero starting vector with a small number of random initial flips before repeating the bit-flipping rounds; this is the same engineering pattern (bounded-attempt decoding with a measured residual failure probability) used in QC-MDPC/BIKE-style decoders [26], [6], and Section V reports the resulting empirical decoding failure rate directly rather than assuming zero. On success, the permutation is undone, $\mathbf{e}_i = \pi^{-1}(\mathbf{v}_i)$, and the plaintext block is recovered by colex-ranking the support of $\mathbf{e}_i$ back to an integer and then to bits.

H. Complete Algorithm

Algorithms 1–5 compose into the full pipeline of Fig. 1: KEYGEN $\rightarrow$ ENCRYPT $\rightarrow$ DCTEMBED $\rightarrow$ (channel) $\rightarrow$ DCTEXTRACT $\rightarrow$ DECRYPT. Table I summarizes the asymptotic cost of each stage as implemented (Section VII): the reference implementation represents H and H_{pub} as dense TABLE I: Per-stage asymptotic complexity ($n=2L$, $r=L$, B = number of blocks, β = carriers/block, R = decode restarts, I = bit-flip rounds/attempt, HW = image pixels).

Stage	Complexity (as implemented)
Key generation	$O(r^3)$ (dense inversion + H_{pub} product)
Encryption	$O(B \cdot rn)$
DCT embedding	$O(HW)$ ($O(1)$ per 8×8 block, fixed $N=8$)
DCT extraction	$O(HW)$
Decryption (per block)	$O(R \cdot I \cdot rn)$ dense; $O(R \cdot I \cdot wn)$ sparse

NumPy arrays for clarity and ease of verification, so matrix–vector products cost $O(rn)$ rather than the $O(wn)$ a sparsityexploiting implementation of H (not H_{pub} , which is deliberately dense) would achieve; this is stated explicitly here and revisited as an optimization opportunity in Section ??.

Two independent error sources compound across the pipeline and are measured separately in Section V: (i) the DCT channel bit-error rate, introduced by the round/clip step of Section IV-E and by any transmission-channel processing (e.g., JPEG re-compression) between embedding and extraction; and (ii) the QC-LDPC decoding-failure rate of Algorithm 5, intrinsic to bit-flipping decoding of the private code regardless of channel quality. A production deployment would typically add an outer error-detecting code (e.g., a CRC per block) so that a receiver can distinguish “block decoded to the wrong plaintext” from “block decoding failed,” rather than silently returning an incorrect block; the reference implementation instead reports both failure classes directly as measured quantities (Section V) and does not add outer redundancy, which is stated as a scope limitation in Section ??.

V. EXPERIMENTAL RESULTS

A. Experimental Setup

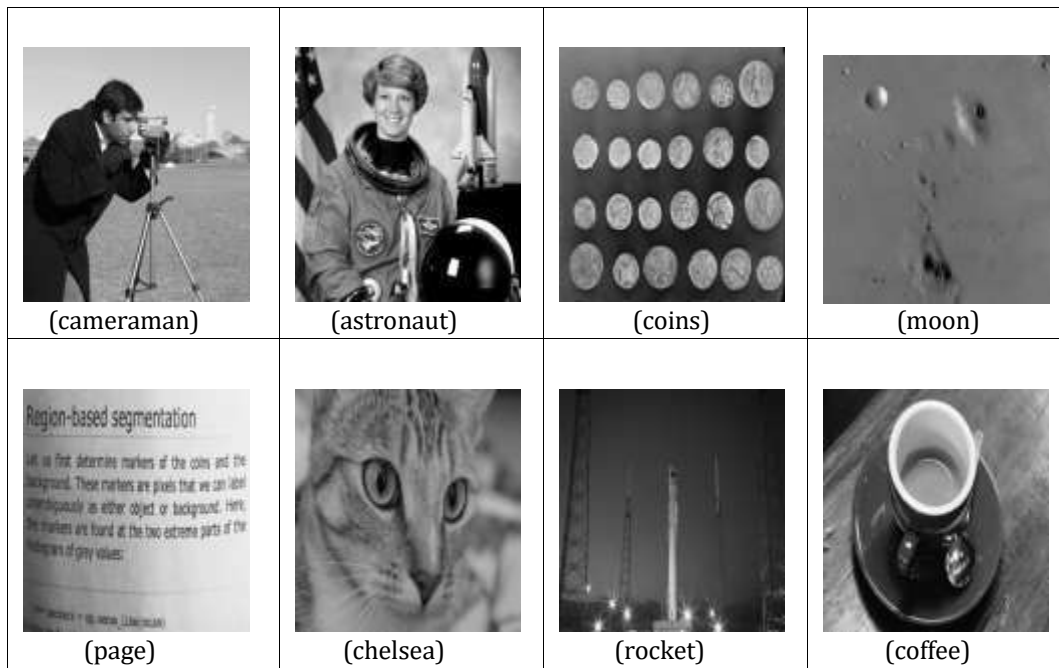
All experiments use the parameter set $(L, w, t) = (96, 7, 4)$, giving $n = 192$, $r = 96$, $k = 96$ (design rate $k/n = 0.5$) and a per-block plaintext capacity of $\lfloor \log_2 \binom{192}{4} \rfloor = 25$ bits

(Section III-B); this set was chosen, from a parameter sweep reported in the released experiment scripts, as the smallest tested configuration attaining a 100% bit-flipping decoding success rate over 100 independent trials before the larger, precision run below. DCT embedding uses quantization step $\Delta = 8$ unless otherwise stated. The cover-image set is eight standard 512×512 , 8-bit grayscale test images (TABLE II) bundled with scikit-image (cameraman, astronaut, coins, moon, page, chelsea, rocket, coffee), covering a range of content types (portrait, texture, low-contrast, line-art, natural scene). All experiments were executed on a single CPU core with no GPU acceleration, using the Python implementation of Section VII; every number and figure in this section is produced directly by the released experiment scripts (run_experiments.py, make_figures.py) with no manual editing.

B. Key Generation Reliability and Cost

Over 200 independent key-generation trials, mean keygeneration time was 23.4 ± 13.7 ms (Fig. 8); the spread reflects

TABLE II: Test cover images (512×512 pixels)



the rejection-sampling loop for a nonsingular S , whose trial count is a small-mean geometric random variable. Paired with each trial, a 64-byte message (21 blocks) was encrypted and decrypted to accumulate an empirical block-level decoding failure rate (DFR): across 4200 decoded blocks, 4 failed to converge within the iteration/restart budget of Algorithm 5, giving $\text{DFR} = 4/4200 = 9.52 \times 10^{-4}$ per block. This is measured, not assumed, and is reported here in the same spirit as QC-MDPC/BIKE's own DFR characterization [26], [6]: because a B-block message fails end-to-end whenever any of its B blocks fails, the probability of at least one failure grows with message length as $1 - (1 - \text{DFR})^B$, e.g., $\approx 16\%$ at $B = 170$ blocks (the largest configuration in Section V-C below) – consistent with the 47/64 exact end-to-end message matches observed in the main sweep (Section V-C).

C. Payload Sweep: Image Quality vs. Capacity

Fig. 2–4 report PSNR, SSIM, and MSE for all 8 images at

8 embedding densities $\beta \in \{1, 2, 3, 4, 6, 8, 12, 16\}$ bits/block TABLE III: Cross-image mean PSNR/SSIM/MSE and perstage timing vs. embedding density β (bits/block, $\Delta=8$, $n=8$ images/point).

β Cap. (bits)	PSNR (dB)	SSIM	MSE	KeyGen (ms)	Enc. (ms)	Embed (ms)	Extract (ms)	
1	4096	50.96	0.995	0.52	18.2	1.11	66.3	37.1
2	8192	47.77	0.990	1.09	25.5	2.11	72.9	43.0
3	12288	46.10	0.985	1.60	24.7	3.11	78.4	48.5
4	16384	44.93	0.981	2.09	25.2	4.14	82.9	53.3
6	24576	43.20	0.972	3.12	24.6	6.14	97.0	64.5
8	32768	42.00	0.963	4.10	25.6	8.25	109.5	75.1
12	49152	40.30	0.947	6.07	28.0	12.65	133.0	96.2
16	65536	39.06	0.931	8.07	26.5	16.95	158.3	120.3

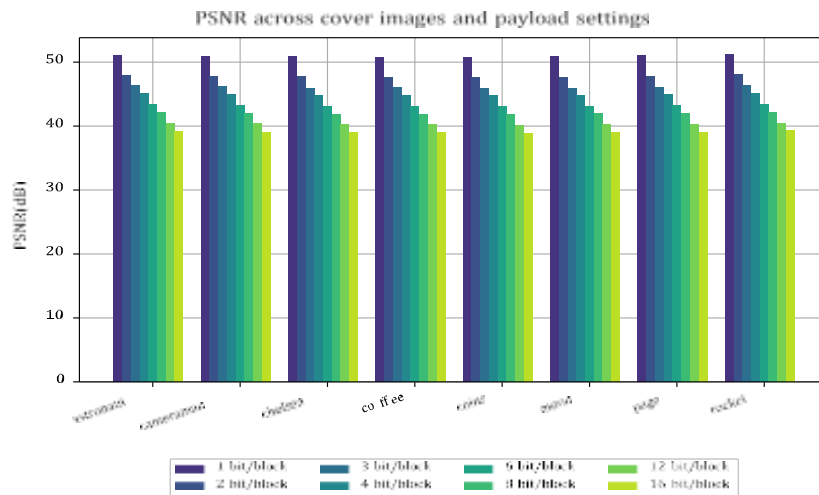


Fig. 2: PSNR across the 8 cover images and all 8 embedding densities $\beta \in \{1,2,3,4,6,8,12,16\}$ (numeric values in Table III). PSNR varies by at most 0.9 dB across images at fixed β , indicating the embedding cost is dominated by β , not image content.

($\Delta = 8$), i.e., 64 independent runs. Table III gives the corresponding cross-image means. PSNR falls monotonically from 50.96 ± 0.16 dB at $\beta = 1$ (4096-bit capacity) to 39.06 ± 0.12 dB at $\beta = 16$ (65536-bit capacity), and SSIM from 0.995 to 0.931, matching the expected capacity/imperceptibility tradeoff of Section III-E: every additional carrier coefficient per block perturbs one more mid-frequency DCT term, so cumulative distortion grows roughly linearly with β while PSNR (a logarithmic measure) declines correspondingly. All 64 configurations remain in the ≥ 39 dB range conventionally treated as “imperceptible” in the steganography literature. Fig. 5 plots embedding capacity directly against β via Eq. (7): capacity is exactly linear in β by construction (4096 bits per unit of β for a 512×512 cover), so this figure is a design-time planning aid rather than a measured trade-off. Fig. 6 and Fig. 7 replot the same PSNR/SSIM data directly against embedded payload size (bits, averaged over the 8 images with error bars showing cross-image standard deviation), giving the practical operating curve a system designer would use to pick β for a target payload size and quality floor.

D. Timing Analysis

Fig. 8 shows the distribution of key-generation time across all 64 main-sweep runs (median close to the 23.4 ms trial mean reported above, with a right tail from occasional additional SSIM across cover images and payload settings

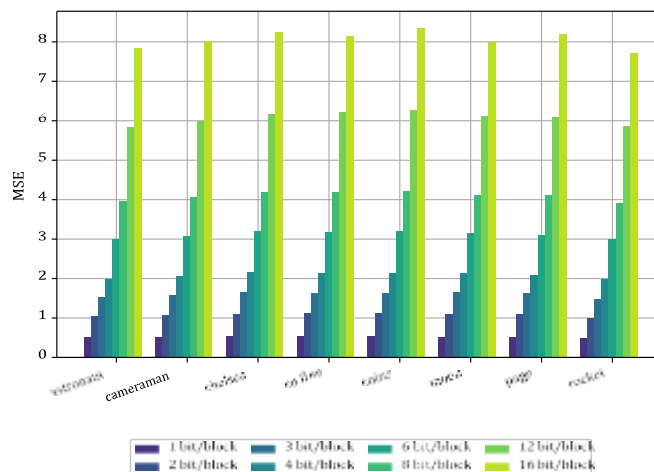


Fig. 3: SSIM across the 8 cover images and embedding densities, mirroring the PSNR trend of Fig. 2. MSE across cover images and payload settings

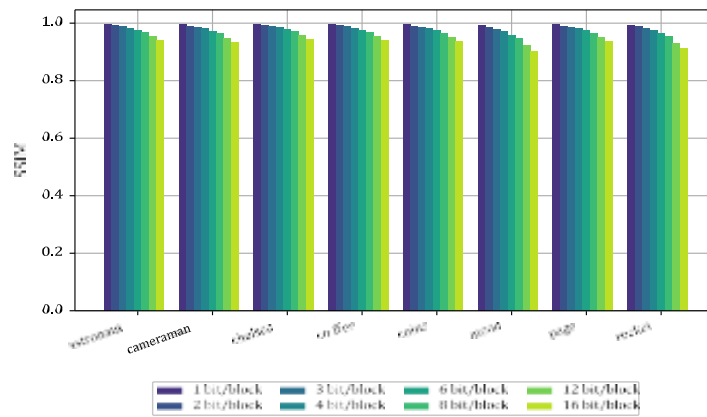


Fig. 4: MSE across the 8 cover images and embedding densities. MSE grows approximately linearly with β , as expected from a fixed per-coefficient perturbation magnitude applied to a linearly growing number of coefficients.

rejection-sampling rounds for S). Fig. 9 plots mean encryption and decryption time against the number of Niederreiter blocks in the message: encryption is linear and fast (1.1–16.9 ms across the sweep, Table III), since it is a single dense matrix– vector product per block, whereas decryption is linear but two to three orders of magnitude slower (312–10958 ms), dominated by the iterative bit-flipping rounds and occasional restart attempts of Algorithm 5 – consistent with Table I’s complexity analysis, and the clearest quantitative illustration in this paper of the dense-vs-sparse implementation gap discussed in Section ?? Fig. 10 shows DCT embedding and extraction time against β : both grow slowly and near-linearly (66–158 ms embed, 37–120 ms extract, Table III), reflecting the fixed per-block 8×8 DCT/IDCT cost times a growing number of touched blocks, and remain two to three orders of magnitude faster than decryption throughout, so decryption

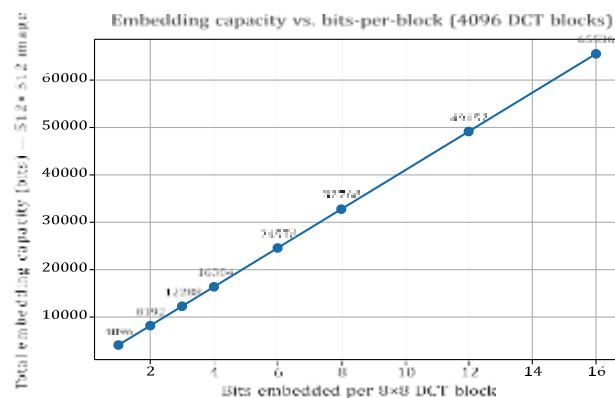


Fig. 5: Embedding capacity (Eq. (7)) vs. carriers/block β for a 512×512 cover image; exactly linear by construction.

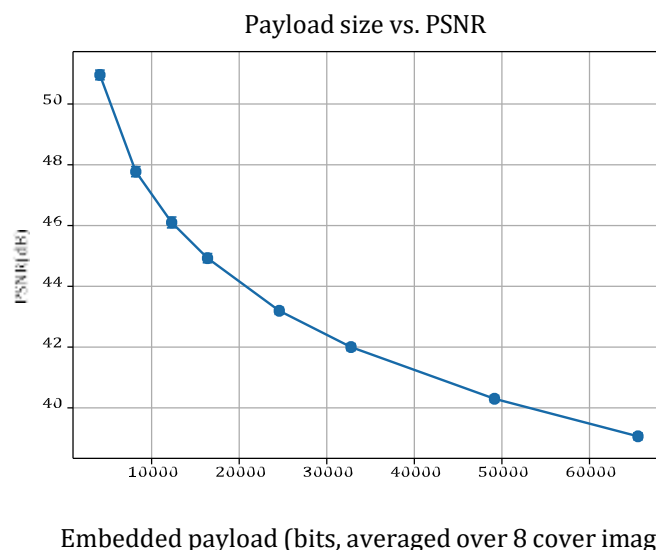
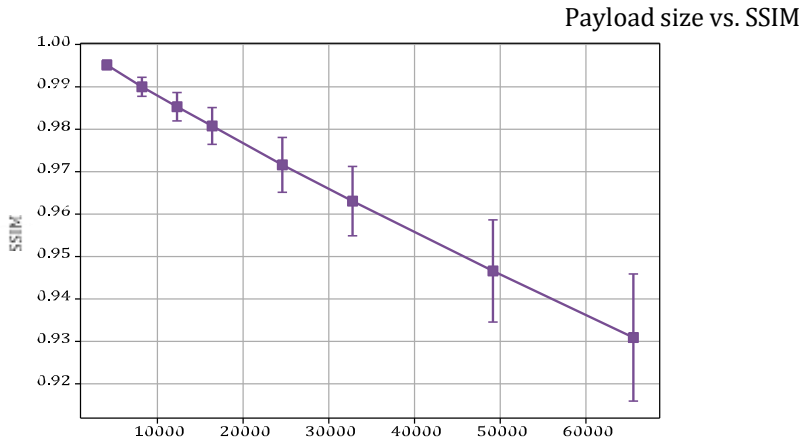


Fig. 6: Payload size (bits, averaged over 8 images, error bars = cross-image std. dev.) vs. PSNR.

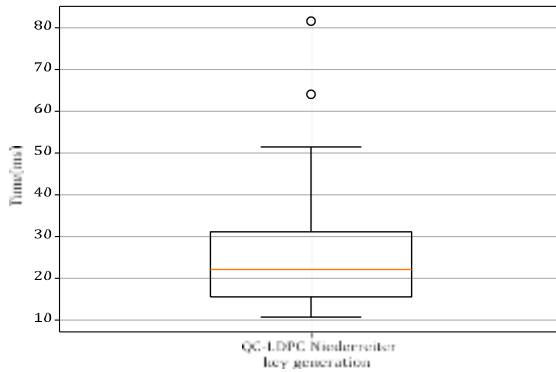
– not the steganographic layer – is the pipeline’s throughput bottleneck at these parameters.

E. DCT Channel Bit-Error Rate

Because both encryption and the DCT channel are deterministic modulo the round/clip step of Section IV-E, the channel bit-error rate (BER between embedded and extracted ciphertext bits, with no transmission attack) is 0 in 51 of the 64 sweep configurations. In the remaining 13, BER is small (mean 1.42×10^{-3} , max 1.63×10^{-2}) and is concentrated in one cover image (astronaut, BER between 8.2×10^{-4} and 1.63×10^{-2} across all 8 densities), indicating that this effect is driven by cover-image content specifically, large smooth regions whose DCT coefficients sit close to a quantization decision boundary in Eq. (6), where the round/clip step can tip the reconstructed coefficient to the wrong side of the boundary rather than by β or by the cryptographic layer. This channel BER, though small, is not zero, and because neither AES-CTR nor the Niederreiter layer used here includes an outer error-detecting/correcting code, a single such bit flip is sufficient to



Embedded payload (bits, averaged over 8 cover images)
Fig. 7: Payload size (bits, averaged over 8 images) vs. SSIM.



Key generation time (n=64 runs, mean=23.4 ms)

Fig. 8: Key-generation time distribution (64 runs).

F. Robustness to JPEG Re-Compression

Fig. 11 plots extraction BER against JPEG re-compression quality factor for the proposed DCT embedder at $\Delta \in \{8, 16, 24\}$ and for the LSB baseline, on the cameraman image. At $\Delta = 8$, BER is 0 at quality 100, rises to 4.5×10^{-3} at quality 80, and approaches the random-guessing level (BER ≈ 0.5) only below quality 50; larger Δ (16, 24) trades imperceptibility for a flatter, lower BER curve at low quality, as expected since a larger quantization step is harder for JPEG’s own quantization to perturb across a decision boundary. In contrast, spatial LSB embedding already exhibits 9.2–9.4% BER at quality 100 (JPEG’s DCT-domain quantization corrupts spatial LSBs even at minimal compression) and reaches the random-guessing regime by quality 90. This is the expected, literature-consistent result that DCT-domain embedding is materially more compression-robust than spatial LSB embedding, though neither survives aggressive (quality ≤ 50) re-compression, which both methods are not designed to resist; the proposed scheme is therefore appropriate for lossless (e.g., PNG/BMP) transport channels or mildly-recompressed channels, not for pipelines that apply

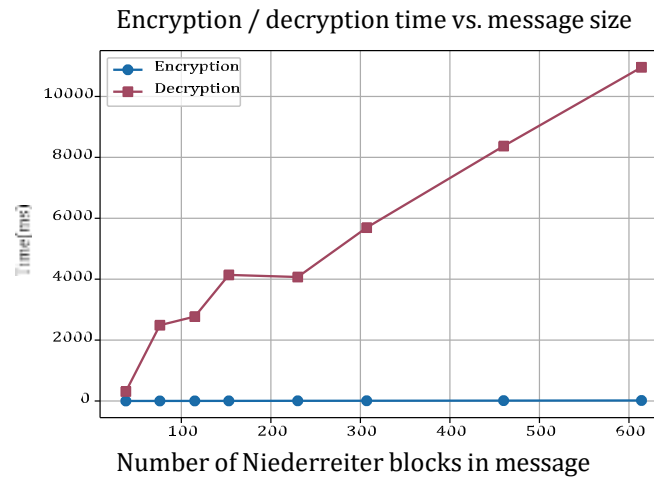


Fig. 9: Mean encryption/decryption time vs. number of Niederreiter blocks in the message.

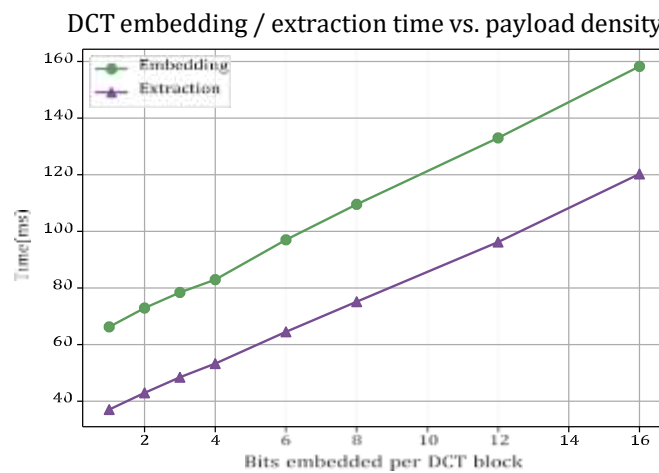


Fig. 10: Mean DCT embedding/extraction time vs. bits embedded per block β .

G. Baseline Comparison

Table IV and Fig. 12 compare four methods under an equal embedding-bit budget: each method embeds exactly (or, for the LSB spatial-domain method, up to) target = 12288 bits into each of the 8 cover images ($\beta = 3$, $\Delta = 8$ for the two DCTdomain crypto methods and for plain DCT). This equal-budget design is deliberate: an earlier, unequalized comparison (in which each method's ciphertext/plaintext was embedded at its own natural length) showed the proposed method embedding 3.84× more bits than AES+DCT for the same plaintext an artifact of Niederreiter's syndrome ciphertext expansion (Section IV), not a property of the embedding domain which would have confounded a quality comparison; Table IV isolates the embedding-domain effect from this expansion effect by fixing the bit budget and instead reporting the plaintext bytes delivered in that budget as its own column.

TABLE IV: Equal-bit-budget baseline comparison (target = 12288 embedded bits, mean over 8 cover images). "Plaintext" is the useful secret payload delivered in that fixed bit budget.

Method	PSNR (dB)	SSIM	Plaintext (bytes)	Embed (ms)	Extract (ms)	Encrypt (ms)	Decrypt (ms)
Proposed (Niederreiter+DCT)	45.64	0.9838	400	87.4	53.7	3.49	3629.3
AES-256-CTR+DCT	45.64	0.9838	1536	88.3	52.7	0.58	0.07
Plain DCT (no crypto)	45.72	0.9844	1536	86.7	52.9	–	–
LSB (no crypto)	64.43	0.9997	1536	0.19	0.04	–	–

Extraction BER vs. JPEG re-compression (cameraman, 512×512)

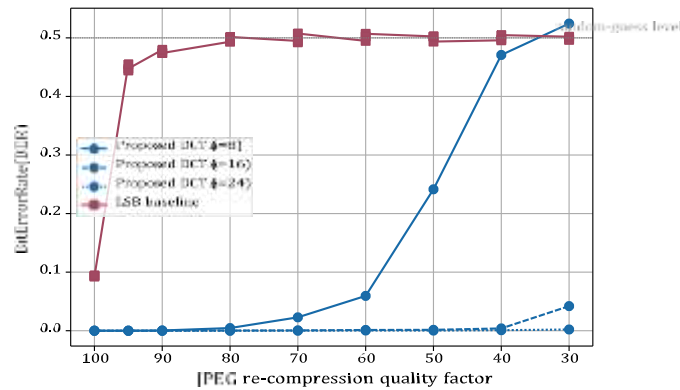


Fig. 11: Extraction bit-error rate vs. JPEG re-compression quality factor (cameraman, 512×512). Proposed DCT embedding at three quantization steps Δ vs. spatial LSB embedding. Baseline comparison: Proposed vs. AES+DCT vs. Plain DCT vs. LSB (avg. over 8 images)

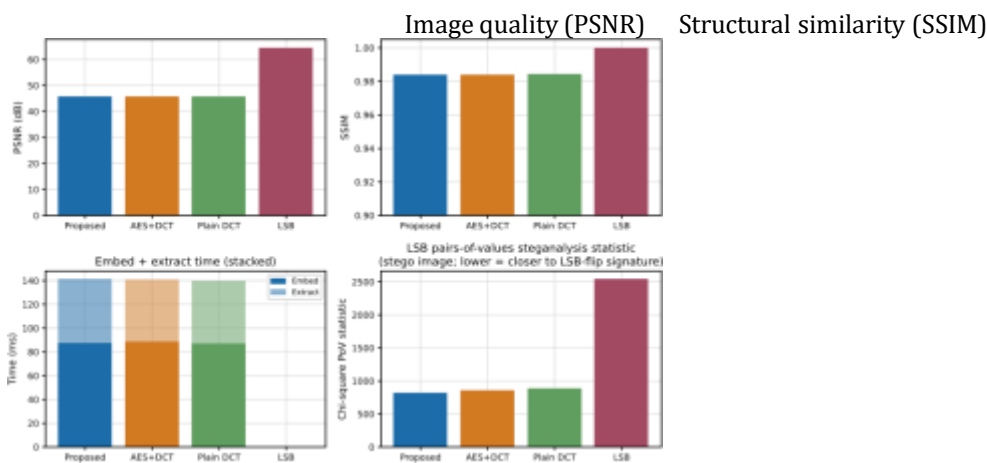


Fig. 12: Baseline comparison at equal embedding-bit budget: PSNR, SSIM, embed+extract time, and the chi-square pair-of-values steganalysis statistic (Section VI), averaged over 8 cover images.

Three findings follow directly from Table IV. First, PSNR/SSIM are statistically indistinguishable across the three DCT-domain methods (45.64-45.72 dB, 0.9838–0.9844): once the bit budget is equalized, the choice of cryptographic layer (Niederreiter, AES, or no encryption at all) has no measurable effect on stego-image fidelity, because the DCT embedder treats its input as an opaque bitstream (Section IV-A). Second, the Niederreiter layer delivers only 400 of the 1536 plaintext bytes an unencrypted or AES-encrypted payload delivers in the same bit budget – a 3.84× rate cost that is the direct, measured consequence of the code’s rate $\text{cap}/r = 25/96 = 0.260$ (Section IV); this is an honest limitation of the specific QCLDPC parameters chosen here, not of code-based cryptography in general (Section ?? discusses parameter choices that reduce this cost). Third, spatial LSB embedding attains substantially higher fidelity (64.4 dB PSNR, SSIM 0.9997) at negligible embedding/extraction cost (sub-millisecond), which is expected since a single-bit perturbation of pixel intensity is far smaller than a DCT-coefficient quantization step – but Section VI shows this fidelity comes at a steganalytic detectability cost the DCT-domain methods do not pay.

VI. SECURITY ANALYSIS

This section separates theoretical security arguments (what the underlying hardness assumptions imply, and at what parameter scale) from measured statistical/steganalytic indicators (Section V’s methodology applied to security-relevant statistics). Neither category supports a claim of unconditional or “unbreakable” security, and none is made.

A. Key-Space Size and the Demonstration-vs-Secure Parameter Gap

Two distinct quantities are relevant and must not be conflated. First, the masking key space – the number of distinct (S,P) pairs an attacker without structural insight would need to search – is $\log_2 \left(\prod_{i=0}^{r-1} (2^r - 2^i) \cdot n! \right) \approx 10398.6$ bits at $r = 96, n = 192$; this is astronomically large but is not the relevant security parameter, because S and P are not directly attacked – an adversary attacks the public code H_{pub} itself via decoding, for which (S,P) provide no additional entropy beyond hiding H ’s structure (Section III-C). Second, and this is the relevant quantity: the number of candidate

weight-t error vectors is $\binom{n}{t} = \binom{192}{4} \approx 3.54 \times 10^7$, i.e., $\log_2 \binom{n}{t} = 25.71$

bits. At the demonstration parameters used throughout this paper, this space is small enough for exhaustive search on commodity hardware in well under a second, i.e., these parameters provide no cryptographic security and are used exclusively to obtain a fully executable, empirically-verifiable prototype (Sections IV–V). Realistic, standardization-scale code-based parameter sets targeting 128-bit classical/post-quantum security use code lengths on the order of $n \sim 10^4$ – 10^5 with information-set-decoding (ISD) work factors calibrated using the Stern/BJMM-family estimation methodology of Peters and others [14], [16], [17], as instantiated by the NIST PQC candidates Classic McEliece and BIKE [7], [6]; scaling the present construction to such n is a direct parameter change to Algorithm 1 (larger L , larger t) with no change to the algorithmic structure, but it was not performed here because it is computationally unnecessary for validating the pipeline’s correctness and interaction behavior, which is this paper’s purpose (Section ?? elaborates).

B. Structural Assumption and QC-LDPC-Specific Caveats

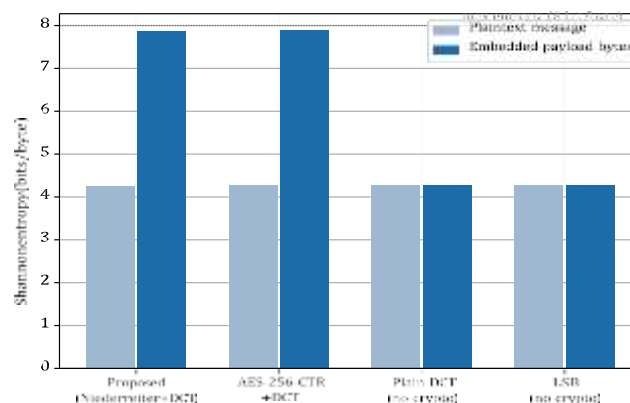
Beyond generic ISD hardness (which applies to any linear code), quasi-cyclic constructions must additionally resist structural distinguishing attacks that exploit the circulant description itself; this is an active research question for QCMDPC/QC-LDPC-family codes and is precisely why standardized instantiations (e.g., BIKE) specify not just (n, t) but the exact circulant weight w , decoder, and a validated, published decoding-failure-rate (DFR) analysis, since a nonnegligible DFR combined with decryption-oracle access enables reaction-type key-recovery attacks [26], [6]. This paper reports its own empirical DFR (9.52×10^{-4} /block, Section V) as a transparency measure, not as a claim of adequacy for any deployment threat model; at deployment scale, DFR must be driven low enough (typically targeted below $2^{-\lambda}$ for security parameter λ) that it does not open a reaction-attack side channel, which requires a decoder and parameter analysis beyond this paper’s scope.

C. Post-Quantum Relevance

The security argument for the cryptographic layer rests on syndrome decoding of a linear code being hard for both classical and (to within the polynomial Grover-type speedup applicable to unstructured search) quantum adversaries, in contrast to the factoring/discrete-log assumptions broken by Shor’s algorithm [1], [2]. This is the entire post-quantum motivation for choosing a code-based cipher as the encryption layer beneath the steganographic channel (Section I); no claim is made that DCT-domain steganography itself carries any post-quantum property, since concealment and computational hardness are different security goals (Section I).

D. Entropy Analysis

Fig. 13 reports the Shannon entropy (bits/byte) of the plaintext message and of the actual embedded payload bytes, for each method of Table IV, averaged over the 8 cover images. The plaintext (natural-language text, Section V) has entropy 4.25–4.27 bits/byte, well below the 8-bit/byte maximum. After Niederreiter encryption the embedded payload’s entropy rises to 7.87 bits/byte, and after AES-256-CTR encryption to 7.88 bits/byte – both close to the theoretical maximum and statistically indistinguishable from one another – while the unencrypted (Plain DCT and LSB) methods embed the plaintext’s own 4.25–4.27 bits/byte unchanged. This is the intended, measured effect of the cryptographic layer: it does not change stego-image fidelity (Section V–G) but it does change what a steganalyst would observe in the payload if they successfully extracted it, from recognizable low-entropy structure to near-uniform noise, for either cipher choice.



Payload entropy: plaintext vs. embedded content (avg. over 8 images)

Fig. 13: Shannon entropy of the plaintext message vs. the embedded payload bytes, by method (mean over 8 images). Adjacent-pixel correlation: cover vs. stego (proposed method)

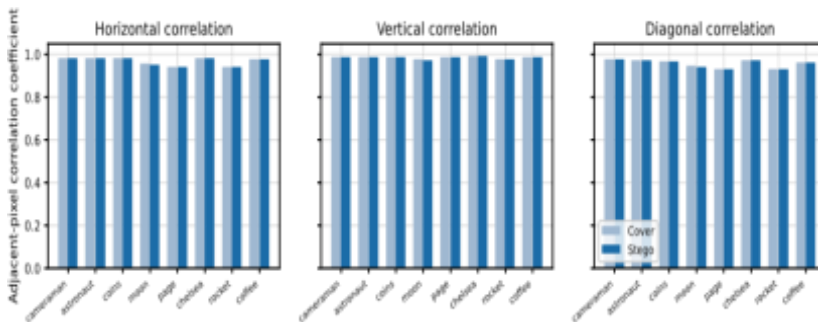


Fig. 14: Horizontal/vertical/diagonal adjacent-pixel correlation, cover vs. stego (proposed method, all 8 images).

E. Pixel Correlation Analysis

Adjacent-pixel correlation (Eq. analogous to a Pearson correlation over horizontally/vertically/diagonally adjacent pixel pairs, 3000 sampled pairs/image) is a standard indicator of whether an image retains natural spatial structure or has been driven toward noise-like statistics [35]. Fig. 14 compares cover and stego correlation (proposed method, $\beta = 3$) across all 8 images and all three directions. Cover-image correlation ranges 0.93–0.98 (horizontal), 0.97–0.99 (vertical), and 0.92–0.98 (diagonal); stego-image correlation tracks the corresponding cover value to within 0.001–0.002 in every case. This is the expected result for an imperceptible embedding at $\Delta = 8, \beta = 3$: the DCT-domain perturbation is far too small, relative to natural-image spatial correlation, to be visible in this statistic, which is also why the values are uninformative for detecting the embedding on their own (a steganalyst would need a targeted, calibrated attack, Section II).

F. Histogram Analysis

Fig. 15 shows the pixel-intensity histogram of the cameraman cover image and its proposed-method stego counterpart. The two histograms are visually near-identical, consistent with the correlation and entropy results above: because embedding perturbs frequency-domain coefficients by at most $\pm\Delta/2$ (Eq. (6)) and the corresponding spatial-domain change is spread by the inverse DCT across all 64 pixels of the block, no single spatial-domain intensity bin is driven toward a detectable spike or dip, unlike the characteristic “comb” lower PoV score is evidence the DCT-domain methods resist the specific, classical attack better than LSB.

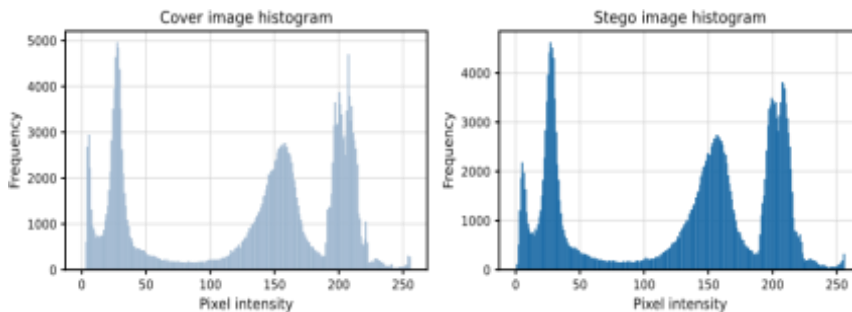


Fig. 15: Pixel-intensity histogram, cover vs. stego (cameraman, proposed method).

G. Differential Analysis (NPCR/UACI)

The number-of-pixels-change rate (NPCR) and unified average changing intensity (UACI), standard differential-analysis statistics from the image-encryption literature [43], are repurposed here to quantify how much the stego image differs pixelby-pixel from its cover. For the proposed method (Table IV’s configuration), NPCR = 67.75% and UACI = 0.388%, essentially identical to AES+DCT (67.74%, 0.388%) and Plain DCT (67.26%, 0.383%), and starkly different from LSB (NPCR = 2.34%, UACI = 0.0092%). This is a direct, measured consequence of the embedding domain, not the cryptographic layer: DCT embedding changes roughly two-thirds of spatial pixels by a small amount each (because the inverse DCT spreads each coefficient perturbation across the full 8×8 block), whereas LSB changes only the small fraction of pixels selected for embedding, each by exactly ± 1 . High NPCR with low UACI is therefore a signature of transform-domain embedding specifically, not an indicator of weak or strong encryption, and should not be over-interpreted as a security metric on its own; it is reported here for completeness because NPCR/UACI are commonly requested in this literature, with this caveat made explicit.

H. Statistical Steganalysis Resistance (Chi-Square Pairs-of-Values)

The Westfeld–Pfitzmann chi-square pairs-of-values (PoV) statistic [39] is computed on each stego image’s pixel histogram (Section II) as a comparable, well-established detectability proxy across methods; lower values indicate a histogram closer to the cover’s natural (non-PoV-equalized) distribution. Table IV’s methods score 820.7 (proposed),

853.0 (AES+DCT), 884.0 (Plain DCT), and 2537.7 (LSB), mean over 8 images (Fig. 12, bottom-right panel). The three DCTdomain methods are statistically similar and all substantially lower than LSB's score, which is the expected result: LSB's odd/even bit-plane overwrite directly targets the value-pair structure the PoV statistic measures, while DCT-domain embedding perturbs a transform coefficient that is spread, by the inverse transform, across many spatial-domain values, diluting the PoV signature. This result should be read narrowly: the PoV statistic is calibrated to LSB-style embedding, and a lower PoV score is evidence the DCT-domain methods resist this specific, classical attack better than LSB, not evidence of general steganalytic undetectability against calibrated or learning-based DCT steganalysis, which is outside this paper's scope.

I. Cryptographic Security Assumptions and Limitations, Summarized

To avoid any ambiguity: the security of the cryptographic layer, at secure parameter scale, rests on the conjectured average-case hardness of bounded-distance syndrome decoding for structured quasi-cyclic codes (Sections III-C-VI-A), a standard but not information-theoretic assumption, shared with the entire code-based PQC family [12]. At the demonstration parameters actually used in this paper's experiments, that assumption is moot because exhaustive search is feasible (Section VI-A); no security claim is made for the executed prototype beyond functional correctness. The steganographic layer's "security" is concealment, not confidentiality, and is bounded by the specific, classical PoV attack measured in Section VI-H, not by any broader claim of undetectability. Neither layer is claimed to be unbreakable, unconditionally secure, or immune to attacks not evaluated in this paper.

VII. PYTHON IMPLEMENTATION

The complete system of Section IV is implemented in Python 3.14 using NumPy 2.5 for F2 linear algebra, SciPy 1.18 (scipy.fft.dctn/idctn) for the block DCT/IDCT, scikit-image 0.26 for image I/O and SSIM, and PyCryptodome for the AES-256-CTR baseline; no GPU or specialized hardware is required. The codebase is organized as follows.

- src/gf2.py – F2 primitives: circulant construction, Gauss Jordan matrix inversion, rejection-sampled random nonsingular matrices, random permutations, matrix vector products.
- src/combinatorics.py – colex ranking/unranking of t-subsets of $\{0, \dots, n-1\}$ (Section III-B), implementing the bijection between plaintext bit-blocks and weight error vectors.
- src/niederreiter_qcldpc.py – KEYGEN, ENCRYPT, DECRYPT (Algorithms 1, 2, 5), including the multi-restart Gallager-B-style bit-flipping decoder.
- src/dct_stego.py DCTEMBED/DCTEXTRACT (Algorithms 3, 4), the fixed mid-band zig-zag carrier table, and the image-capacity formula (Eq. (7)).
- src/baselines.py LSB embedding/extraction and AES-256-CTR encryption/decryption, used for the comparisons of Section V-G.
- src/metrics.py PSNR/SSIM/MSE/BER, Shannon entropy, adjacent-pixel correlation, the chi-square pairsof-values statistic, NPCR/UACI, and the theoretical masking key-space formula of Section VI-A.
- experiments/ prepare_images.py (coverimage set), run_experiments.py (all measurements of Section V, seeded for reproducibility), make_figures.py (every figure in this paper, generated directly from the CSV/JSON result files), summarize_results.py (the numeric summary used to write Sections V-VI).

Listing 1 shows the core odd/even quantization-index embedding step (Eq. (6)), and Listing 2 shows the syndromedomain bit-flipping decoding round (Section IV-G), reproduced verbatim from the released source.

Listing 1: QIM embedding of one bit into one DCT coefficient (dct_stego.py).

```
q = np.round(c / delta) if bit == 1 and (int(q) % 2 == 0): q += 1
elif bit == 0 and (int(q) % 2 != 0):
    q -= 1
coeffs[ry, rx] = q * delta
```

Listing 2: One bit-flipping decode round against a target syndrome (niederreiter_qcldpc.py).

```
cur = matvec_gf2(H, e).astype(np.uint8) diff = cur ^ target if not diff.any():
    return e if hamming_weight(e) == t else None
unsat = np.array([diff[cs].sum() for cs in col_supports]) max_unsat = unsat.max() flip_idx =
np.nonzero(unsat == max_unsat)[0] e[flip_idx] ^= 1
```

VIII. CONCLUSION

This paper presented a hybrid system combining a quasicyclic LDPC instantiation of the Niederreiter cryptosystem with block-DCT image steganography, specified the exact mathematical interface between the two layers,

implemented every stage as reproducible Python code, and evaluated it empirically not theoretically on eight standard test images. The central results are that (i) the cryptographic layer's choice has no measurable effect on stego-image fidelity once the embedding-bit budget is equalized across methods (45.6– 45.7 dB PSNR for Niederreiter+DCT, AES+DCT, and unencrypted DCT alike); (ii) it does raise the embedded payload's entropy from natural-language levels (~ 4.25 bits/byte) to nearuniform (~ 7.87 bits/byte), matching AES's own effect; (iii) DCT-domain embedding resists JPEG re-compression and the classical chi-square pairs-of-values steganalysis attack substantially better than spatial LSB embedding, at a fidelity cost (45.6 vs. 64.4 dB PSNR); and (iv) the QC-LDPC Niederreiter layer's principal cost is not fidelity but a measured $3.84\times$ ciphertext-expansion rate and a nonzero, empirically characterized decoding-failure rate (9.52×10^{-4} /block), both reported directly rather than glossed over. The demonstration parameters used are explicitly, repeatedly stated to be insecure and intended only to validate the pipeline's correctness and interaction behavior; scaling to standardization-grade parameters is architecturally direct but was left to future work, alongside an outer error-detecting code, a sparse decoder implementation, and evaluation against modern steganalysis.

REFERENCES

- [1] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.
- [2] D. J. Bernstein and T. Lange, "Post-quantum cryptography," *Nature*, vol. 549, pp. 188–194, 2017.
- [3] R. J. McEliece, "A public-key cryptosystem based on algebraic coding theory," Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, DSN Progress Report 42-44, 1978.
- [4] H. Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory," *Problems of Control and Information Theory*, vol. 15, no. 2, pp. 159–166, 1986.
- [5] Y. X. Li, R. H. Deng, and X. M. Wang, "On the equivalence of McEliece's and Niederreiter's public-key cryptosystems," *IEEE Transactions on Information Theory*, vol. 40, no. 1, pp. 271–273, 1994.
- [6] N. Aragon, P. S. L. M. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Ghosh, S. Gueron, T. Guney, C. Aguilar Melchor, R. Misoczki, E. Persichetti, J. Richter-Brockmann, N. Sendrier, J.-P. Tillich, V. Vasseur, and G. Zemor, "BIKE: Bit flipping key encapsulation – round 4 submission," NIST PostQuantum Cryptography Standardization, 2024.
- [7] G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, Y.-K. Liu, C. Miller, D. Moody, R. Peralta, R. Perlner, A. Robinson, and D. Smith-Tone, "Status report on the third round of the NIST post-quantum cryptography standardization process," National Institute of Standards and Technology (NIST), NISTIR 8413, 2022.
- [8] M. R. Albrecht, D. J. Bernstein, T. Chou, C. Cid, J. Gilcher, T. Lange, V. Maram, I. von Maurich, R. Misoczki, R. Niederhagen, K. G. Paterson, E. Persichetti, C. Peters, P. Schwabe, N. Sendrier, and J. Szefer, "Classic McEliece: Conservative code-based cryptography round 4 submission," NIST Post-Quantum Cryptography Standardization, 2022.
- [9] P. Gaborit, C. Aguilar-Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, E. Persichetti, G. Zemor, J. Bos, A. Dion, J. Lacan, J.-M. Robert, and P. Veron, "Hamming quasi-cyclic (HQC)," NIST Post-Quantum Cryptography Standardization, 2025.
- [10] L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone, "Report on post-quantum cryptography," National Institute of Standards and Technology (NIST), NISTIR 8105, 2016.
- [11] National Institute of Standards and Technology, "Module-lattice-based key-encapsulation mechanism standard," U.S. Department of Commerce, Federal Information Processing Standards Publication FIPS 203, 2024.
- [12] R. Overbeck and N. Sendrier, "Code-based cryptography," in *PostQuantum Cryptography*, D. J. Bernstein, J. Buchmann, and E. Dahmen, Eds. Berlin, Heidelberg: Springer, 2009, pp. 95–145.
- [13] E. Prange, "The use of information sets in decoding cyclic codes," *IRE Transactions on Information Theory*, vol. 8, no. 5, pp. 5–9, 1962.
- [14] J. Stern, "A method for finding codewords of small weight," in *Coding Theory and Applications*, ser. Lecture Notes in Computer Science, vol. 388. Berlin, Heidelberg: Springer, 1989, pp. 106–113.
- [15] D. J. Bernstein, T. Lange, and C. Peters, "Attacking and defending the McEliece cryptosystem," in *Post-Quantum Cryptography – PQCrypto 2008*, ser. Lecture Notes in Computer Science, vol. 5299. Berlin, Heidelberg: Springer, 2008, pp. 31–46.
- [16] A. Becker, A. Joux, A. May, and A. Meurer, "Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding," in *Advances in Cryptology – EUROCRYPT 2012*, ser. Lecture Notes in Computer Science, vol. 7237. Berlin, Heidelberg: Springer, 2012, pp. 520–536.
- [17] C. Peters, "Information-set decoding for linear codes over F_q ," in *Post-Quantum Cryptography – PQCrypto 2010*, ser. Lecture Notes in Computer Science, vol. 6061. Berlin, Heidelberg: Springer, 2010, pp. 81–94.
- [18] P. Gaborit, "Shorter keys for code-based cryptography," in *Proceedings of the 2005 International Workshop on Coding and Cryptography (WCC 2005)*, Bergen, Norway, 2005, pp. 81–91.

- [19] R. G. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.
- [20] *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.
- [21] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 32, no. 18, pp. 1645–1646, 1996.
- [22] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Transactions on Information Theory*, vol. 50, no. 8, pp. 1788–1793, 2004.
- [23] M. Baldi and F. Chiaraluce, "Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes," in *Proceedings of the 2007 IEEE International Symposium on Information Theory (ISIT)*, Nice, France, 2007, pp. 2591–2595.
- [24] A. Otmani, J.-P. Tillich, and L. Dallot, "Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes," *Mathematics in Computer Science*, vol. 3, no. 2, pp. 129–140, 2010.
- [25] M. Baldi, *QC-LDPC Code-Based Cryptography*, ser. SpringerBriefs in Electrical and Computer Engineering. Springer, 2014.
- [26] R. Misoczki, J.-P. Tillich, N. Sendrier, and P. S. L. M. Barreto, "MDPCMcEliece: New McEliece variants from moderate density parity-check codes," in *Proceedings of the 2013 IEEE International Symposium on Information Theory (ISIT)*, Istanbul, Turkey, 2013, pp. 2069–2073.
- [27] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. C-23, no. 1, pp. 90–93, 1974.
- [28] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.
- [29] D. Upham, "Steganographic algorithm JSteg," *Software*, 1993.
- [30] A. Westfeld, "F5—a steganographic algorithm," in *Information Hiding – 4th International Workshop, IH 2001*, ser. Lecture Notes in Computer Science, vol. 2137. Berlin, Heidelberg: Springer, 2001, pp. 289–302.
- [31] C.-K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognition*, vol. 37, no. 3, pp. 469–474, 2004.
- [32] R. Chandramouli and N. Memon, "Analysis of LSB based image steganography techniques," in *Proceedings of the 2001 International Conference on Image Processing (ICIP)*, vol. 3, Thessaloniki, Greece, 2001, pp. 1019–1022.
- [33] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—a survey," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999.
- [34] N. Provos and P. Honeyman, "Hide and seek: An introduction to steganography," *IEEE Security & Privacy*, vol. 1, no. 3, pp. 32–44, 2003.
- [35] A. Cheddad, J. Condell, K. Curran, and P. McKevitt, "Digital image steganography: Survey and analysis of current methods," *Signal Processing*, vol. 90, no. 3, pp. 727–752, 2010.
- [36] I. J. Kadhim, P. Premaratne, P. J. Vial, and B. Halloran, "Comprehensive survey of image steganography: Techniques, evaluations, and trends in future research," *Neurocomputing*, vol. 335, pp. 299–326, 2019.
- [37] J. Fridrich, *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge, UK: Cambridge University Press, 2010.
- [38] C. Fontaine and F. Galand, "How Reed-Solomon codes can improve steganographic schemes," *EURASIP Journal on Information Security*, vol. 2009, p. 274845, 2009.
- [39] A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems," in *Information Hiding – 3rd International Workshop, IH 1999*, ser. Lecture Notes in Computer Science, vol. 1768. Berlin, Heidelberg: Springer, 2000, pp. 61–76.
- [40] J. Fridrich, M. Goljan, and R. Du, "Reliable detection of LSB steganography in color and grayscale images," in *Proceedings of the 2001 ACM Workshop on Multimedia and Security (New Challenges)*, 2001, pp. 27–30.
- [41] J. Fridrich, M. Goljan, and D. Hoge, "Steganalysis of JPEG images: Breaking the F5 algorithm," in *Information Hiding – 5th International Workshop, IH 2002*, ser. Lecture Notes in Computer Science, vol. 2578. Berlin, Heidelberg: Springer, 2003, pp. 310–323.
- [42] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [43] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption," *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT)*, pp. 31–38, 2011. [Online]. Available: <http://www.cyberjournals.com/Papers/Apr2011/05.pdf>
- [44] N. Sendrier, "Encoding information into constant weight words," in *Proceedings of the 2005 IEEE International Symposium on Information Theory (ISIT)*, Adelaide, Australia, 2005, pp. 435–438.
- [45] T. M. Cover, "Enumerative source encoding," *IEEE Transactions on Information Theory*, vol. 19, no. 1, pp. 73–77, 1973.
- [46] N. Sendrier, "Decoding one out of many," in *Post-Quantum Cryptography – PQCrypto 2011*, ser. Lecture Notes in Computer Science, vol. 7071. Springer, 2011, pp. 51–67.