Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Authenticated delivery of BB84-keyed messages through lossy image channels using hardware-derived keys

Arnav Singhal

arnavs22.comp@coeptech.ac.in

COEP Technological University

Aloki Upadhyay

COEP Technological University

Hakimuddin Slatewala

COEP Technological University

Kush Agarwal

COEP Technological University

Mukta Nivelkar

Fr. Conceicao Rodrigues Institute of Technology

Sunil Bhirud

COEP Technological University

Research Article

Keywords: quantum key distribution, BB84, information hiding, JPEG robustness, Reed–Solomon codes, NISQ hardware

Posted Date: November 5th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11624292/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Authenticated delivery of BB84-keyed messages through lossy image channels using hardware-derived keys

Arnav Singhal[1*], Aloki Upadhyay[1], Hakimuddin Slatewala[1], Kush Agarwal[1], Mukta Nivelkar[2*], Sunil Bhirud[1]

[1*]Department of Computer Science and Engineering, COEP Technological University, Shivajinagar, Pune, Maharashtra, India.
[2]Department of Computer Science and Engineering, Fr. Conceicao Rodrigues Institute of Technology, Navi Mumbai, Maharashtra, India.

*Corresponding author(s). E-mail(s): arnavs22.comp@coeptech.ac.in; mukta.nivelkar@fcrit.ac.in;
Contributing authors: alokiu23.comp@coeptech.ac.in; hakimuddins23.comp@coeptech.ac.in; kusha23.comp@coeptech.ac.in; sgbhirud.comp@coeptech.ac.in;

Abstract

Pipelines that key image-based information hiding with quantum key distribution (QKD) assume the stego image arrives bit-exact, and keys from noisy quantum hardware are unusable without reconciliation. This work addresses both. BB84 runs on IBM's ibm_fez are reconciled with Cascade, confirmed by hashing and compressed by Toeplitz privacy amplification; at 4096 qubits every run yields 1,110–1,237 secure bits, identical for both parties, at a quantum bit error rate of at most 0.98%. The keys protect messages with authenticated encryption, Reed–Solomon coding and key-driven embedding in discrete cosine transform (DCT) or wavelet domains. A pre-registered evaluation on 300 images shows that a key-driven least-significant-bit reference scheme, representative of lossless-channel designs, recovers 0 messages after JPEG recompression, while coded DCT embedding delivers at least 99% of messages at every quality tested; restricted post hoc to images above 40 dB peak signal-to-noise ratio, it carries 0.218 bits per pixel at quality 95 and 0.005 at quality 50. A hardware-keyed message decrypts and verifies after JPEG quality 50. This robustness costs detectability: at the smallest payload tested, a simple learned detector separates the robust variants from covers with an area under the curve of at least 0.797, against 0.517 for the

reference scheme. The keys characterize hardware noise rather than deployable secrecy, and the security estimate is asymptotic.

Keywords: quantum key distribution, BB84, information hiding, JPEG robustness, Reed–Solomon codes, NISQ hardware

1 Introduction

Quantum key distribution (QKD) lets two parties establish a shared secret key whose secrecy rests on the laws of quantum mechanics rather than on computational assumptions. The first such protocol, BB84, encodes random bits in randomly chosen conjugate bases of single qubits, so that any interception disturbs the states and raises the error rate. [1] Information hiding addresses a complementary need: carrying a message inside an ordinary-looking cover object, such as a photograph, rather than in a visibly encrypted file. Spatial-domain methods that replace the least significant bits (LSB) of pixels, and transform-domain methods that modify frequency coefficients, are the two main families of image steganography. [2]

Combining the two layers is attractive, and several systems now key an information-hiding pipeline with a QKD-derived key. One such system uses a key from simulated E91 to encrypt the output of a deep-learning image-hiding network. [3] Another hides message bits in the sifted QKD key itself rather than in an image. [4] A third combines a BB84 key with post-quantum key encapsulation and hides the message in a quantum image representation. [5] Spatial LSB embedding is also used directly with QKD keys. [6]

These QKD-keyed steganography schemes assume that the stego image arrives bit-exact. Real image channels do not behave this way: messaging applications and social platforms routinely recompress uploaded images. Authenticated encryption makes the consequence severe. A ciphertext protected by the Advanced Encryption Standard (AES) in Galois/Counter Mode (GCM) is rejected if even one bit is wrong, so a single uncorrected bit error loses the whole message. [7] In the measurements reported here, a representative lossless-channel scheme (key-driven spatial LSB) recovered 0 of 300 messages after JPEG recompression, even at quality 100, the mildest setting tested (Sect. 6).

A second gap concerns the quantum layer. On a noiseless simulator, Alice's and Bob's sifted keys coincide and reconciliation is never exercised, and the hardware BB84 demonstrations found in the literature report sifting and quantum bit error rate (QBER) statistics (Sect. 2.1). On noisy intermediate-scale quantum (NISQ) hardware the sifted keys differ, and a key derived from them is only usable after information reconciliation and privacy amplification.

This paper asks, in quantum terms, what it takes for QKD-derived keys obtained on current NISQ hardware to protect messages delivered over real-world lossy image channels, and what that robustness costs. The contributions are:

- An end-to-end pipeline in which keys from BB84 runs on IBM hardware pass through standard post-processing (Cascade reconciliation and Toeplitz privacy amplification, with secure-key accounting on ibm_fez) before driving the hiding layer.
- Robust QKD-keyed information hiding for lossy channels: key-driven transform-domain embedding with an error-correction layer between authenticated encryption and embedding, so that authenticated decryption survives JPEG recompression and noise.
- A pre-registered, two-stage (pilot and confirmatory) evaluation of spatial, discrete cosine transform (DCT) and discrete wavelet transform (DWT) variants across robustness, capacity, image quality and detectability, including end-to-end runs using a hardware-derived key, showing that robustness to JPEG recompression is obtained at a measurable cost in detectability.

The detectability cost is reported as a main finding: a simple learned detector identifies the transform-domain arms far more easily than the LSB reference scheme. The system is therefore not presented as a means of hiding the presence of a message from a capable steganalyst; it is a robust embedding layer for authenticated, QKD-keyed messages.

Section 2 reviews related work and Sect. 3 the building blocks. Section 4 describes the system and Sect. 5 the experimental protocol. Section 6 reports the results, leading with the quantum layer, Sect. 7 discusses them together with the limitations of the study, and Sect. 8 concludes.

2 Related work

2.1 QKD on cloud quantum hardware

BB84 was proposed as a public key distribution protocol in which eavesdropping is detected through the errors that measurement introduces. [1] Its security against arbitrary attacks was later proved by relating it to an entanglement-purification protocol, which yields an asymptotic key rate that vanishes at a QBER of about 11%. [8] The security of practical implementations, including the roles of error correction, privacy amplification and device imperfections, has been reviewed in detail. [9] Recent work in this journal continues to study how decoherence and other noise sources erode the security of BB84. [10]

Cloud access to superconducting processors has made circuit-level BB84 demonstrations straightforward. The Qiskit software stack provides the circuit construction, simulation and device access that such demonstrations rely on. [11] One recent preprint runs BB84 with and without intercept-resend eavesdropping on `ibm_fez` with 40 qubits per condition, retains 19 and 23 sifted bits, and states that information reconciliation and privacy amplification are not implemented. [12] Another preprint runs BB84 on IBM hardware and reports encoding, measurement, sifting, QBER and key generation rate; its abstract does not mention either reconciliation or privacy amplification. [13] Information reconciliation itself remains an active research topic, with recent work improving its efficiency for high-dimensional QKD. [14] In the present work, keys from hardware runs pass through the complete post-processing chain and are then used to protect messages.

2.2 QKD-keyed steganography pipelines

Sykot et al. generate a key by simulated E91 in Qiskit, hash it with SHA-256 and use it with AES to encrypt the stego image produced by a deep-learning hiding network. Their evaluation covers entropy, the number of pixels change rate (NPCR) and unified average changing intensity (UACI), key-generation rate, timing, histograms and key sensitivity, and the transmitted image is never modified. [3] Rezaeishad and Bahramgiri use the QKD key itself as the cover. One-time-pad-encrypted, Hamming-coded message bits are embedded in the sifted key during post-processing by a block-based search, so the scheme has no image channel at all. [4] Mogos and Lam combine a BB84 key with the ML-KEM key-encapsulation mechanism, encrypt the message with a quantum one-time pad and hide it in a 4×4 NEQR fingerprint image, all in Qiskit simulation. Their attack analysis covers entropy-based steganalysis, amplitude tampering, which a quantum checksum detects but does not correct, and replay, while no compression or channel noise is applied to the cover. [5] Arumugam et al. describe a web tool that combines QKD key exchange with a Caesar cipher and LSB embedding in uploaded images. [6] None of the works examined evaluates delivery through a lossy image channel, or measures authenticated message recovery after recompression.

2.3 Robust steganography for lossy channels

Embedding that survives JPEG recompression is a mature field. Duan et al. select the embedding domain according to the channel and adapt the strength of error correction to the errors that recompression introduces. [15] Zhang, He and Cao design steganographic polar codes that make message extraction error-free after recompression. [16] Many robust schemes build on quantization index modulation (QIM), which embeds information by quantizing host coefficients onto one of several interleaved lattices. [17] Transform-domain embedding combined with error correction is therefore established in this literature and is *not* a contribution of this work, which uses simple, well-understood variants of it. Frameworks that target JPEG robustness and resistance to statistical detection at the same time treat the two goals as competing design constraints. [18] The results reported here quantify that tension for the configurations studied.

2.4 Closest related work

The closest design known to the authors is a QDCT-QIM watermarking scheme that uses a BB84 key stream to select mid-frequency DCT coefficients and the QIM step, and tests robustness to common attacks. [19] It differs from the present work in four respects. It is watermarking, with a fixed pattern per block, rather than secret communication. It reports bit accuracy rather than authenticated recovery of a whole message. Its BB84 keys are simulated rather than obtained on hardware and reconciled. It uses no encryption and no error correction beyond parity. The uncoded DCT-QIM arm of this study is the direct comparison to that design.

2.5 Quantum image steganography

A separate line of work, much of it published in this journal, hides data in images encoded as quantum states. The novel enhanced quantum representation (NEQR) stores the gray value of every pixel in a qubit sequence entangled with its position. [20] LSB-type steganography on NEQR images was proposed shortly afterwards. [21] Later schemes extend it to colour images with turtle-shell embedding. [22] Hash encryption of the secret message before quantum LSB embedding has also been studied. [23] NEQR watermarking has been executed on IBM hardware, at the small image sizes that current devices allow. [24] The idea of the B92 protocol has been applied to the least significant qubits of an NEQR image shared by two parties, producing a random message that can serve as a key. [25] These schemes require quantum image representations and, for practically sized images, fault-tolerant hardware. The system studied here keeps the images classical and uses current quantum hardware only for what it can do today: distributing key material.

3 Preliminaries

3.1 BB84 and classical post-processing

In BB84, Alice prepares N qubits, each encoding a random bit in a randomly chosen rectilinear or diagonal basis, and Bob measures each in a randomly chosen basis. [1] After the bases are announced, positions with matching bases form the sifted keys $\mathbf{a}, \mathbf{b} \in \{0,1\}^{N_\mathrm{s}}$, with $N_\mathrm{s} \approx N/2$.

3.1.1 QBER estimation

A random fraction (25%) of the sifted positions is disclosed and compared. With e errors among m disclosed bits, the point estimate is $\hat{Q} = e/m$. Because the subsequent block sizing must not underestimate the error rate, the one-sided Clopper–Pearson upper bound is also computed,

$$Q^{+} = \mathrm{B}^{-1}(1-\alpha;\, e+1,\, m-e)\,, \qquad \alpha = 0.05, \tag{1}$$

where $\mathrm{B}^{-1}(\cdot\,; p, q)$ is the inverse regularized incomplete beta function. This exact binomial bound is the classical construction of Clopper and Pearson. [26] The run aborts if $\hat{Q}$ exceeds 11%. The disclosed bits are discarded, leaving n bits.

3.1.2 Reconciliation

Cascade corrects Bob's string towards Alice's by exchanging parities over the public channel. [27] In each of 4 passes the key is permuted and split into blocks whose size starts at

$$k_1 = \left\lceil \frac{0.73}{Q^{+}} \right\rceil, \qquad k_{i+1} = 2\,k_i, \tag{2}$$

so that a first-pass block contains an error with probability close to one half. Blocks whose parities differ are bisected until an error is located and flipped, and each correction is propagated back to blocks of earlier passes. Every disclosed parity bit counts towards the reconciliation leakage L_{EC}.

3.1.3 Key confirmation

Cascade can leave an even number of errors undetected in rare cases. Alice and Bob therefore compare a t-bit hash of their reconciled strings ($t = 64$) drawn from a universal family, and abort on mismatch. [28] These t bits are public and are counted as leaked.

3.1.4 Privacy amplification

The reconciled n-bit string $\mathbf{x}$ is compressed to ℓ bits with a random binary Toeplitz matrix $\mathbf{T} \in \{0,1\}^{\ell \times n}$ chosen from a public seed,

$$\mathbf{k} = \mathbf{T}\mathbf{x} \bmod 2. \tag{3}$$

Universal hashing of this kind distils a short, nearly secret key from a longer string about which an adversary holds partial information. [29] Toeplitz matrices form a two-universal family, and the leftover hash lemma bounds the distance of the output from a uniform key independent of the adversary's information E,

$$\frac{1}{2}\,\bigl\|\rho_{\mathbf{k}E} - \tau_\ell \otimes \rho_E\bigr\|_1 \;\le\; \frac{1}{2}\,2^{-(H_{\min}(\mathbf{x}\,|\,E)-\ell)/2}, \tag{4}$$

where τ_ℓ is the uniform distribution on ℓ bits and $H_{\min}$ the conditional min-entropy. [30] In the asymptotic limit, the min-entropy available per sifted bit after an error rate Q leads to the Shor–Preskill key rate

$$r = 1 - 2\,h(Q), \qquad h(p) = -p\log_2 p - (1-p)\log_2(1-p), \tag{5}$$

which falls to zero at $Q \approx 11\%$. [8] Following this accounting, which keeps the adversary's information below the margin required by (4), the length of the final key is estimated as

$$\ell = \max\!\Bigl(0,\; \bigl\lfloor n - L_{\mathrm{EC}} - t - n\,h(\hat{Q}) - s \bigr\rfloor\Bigr), \tag{6}$$

where the reconciliation leakage L_{EC} is subtracted explicitly instead of the second entropy term of (5), and $s = 64$ bits is a fixed margin. Equation (6) is a teaching-level estimate and not a finite-key bound. [31] Its consequences are discussed in Sect. 7.4.

3.2 Key derivation and authenticated encryption

The extract-and-expand key derivation function based on the hash-based message authentication code (HKDF) turns one secret into several independent-looking subkeys by expanding it with distinct context labels. [32] The Advanced Encryption Standard (AES) is the block cipher used to protect the message. [33] In Galois/Counter

Mode (GCM), AES provides confidentiality and integrity with the recommended 12-byte nonce and a 16-byte authentication tag. [7] Decryption either returns the exact plaintext or rejects the input; there is no partial result.

3.3 Quantization index modulation

Dither-modulation QIM embeds a bit $m \in \{0, 1\}$ into a real coefficient c by quantizing it onto one of two interleaved lattices,

$$c' = \Delta \operatorname{round}\left(\frac{c - d_m}{\Delta}\right) + d_m, \qquad d_1 = d_0 + \frac{\Delta}{2}, \tag{7}$$

where Δ is the step size and d_0 is a key-derived dither. [17] The receiver decides for the lattice nearest to the received coefficient. Because the two lattices in (7) are $\Delta/2$ apart, a bit survives any perturbation of magnitude below $\Delta/4$. Larger Δ increases robustness and distortion together.

3.4 Reed–Solomon codes

A Reed–Solomon (RS) code over GF(2^8) with codeword length 255 bytes and n_{sym} parity bytes carries $255 - n_{\text{sym}}$ data bytes. [34] A codeword with e byte errors at unknown positions and s erasures at known positions is decoded correctly whenever

$$2e + s \leq n_{\text{sym}}, \tag{8}$$

so without erasures up to $\lfloor n_{\text{sym}}/2 \rfloor$ byte errors per codeword are corrected. Beyond that limit the decoder either reports failure or decodes to a wrong codeword.

4 System design

4.1 Overview

Figure 1 shows the pipeline. Alice and Bob run BB84 over the quantum layer and post-process the result over an authenticated classical channel, each ending with a privacy-amplified key. Each party derives subkeys from their *own* key. Alice encrypts the message, encodes the ciphertext with an RS code and embeds the coded bits in a cover image. The stego image passes through a lossy cover channel. Bob extracts the bits, decodes them and attempts authenticated decryption.

4.2 Quantum layer

4.2.1 Circuits

Each BB84 position is a single-qubit circuit: Alice's bit sets an X gate, her basis an H gate, and Bob's basis a further H before measurement. On hardware, 100 positions are packed side by side into one wide circuit on the device's lowest-readout-error qubits, and all circuits of a run are submitted as one job. This reduces queue time but means

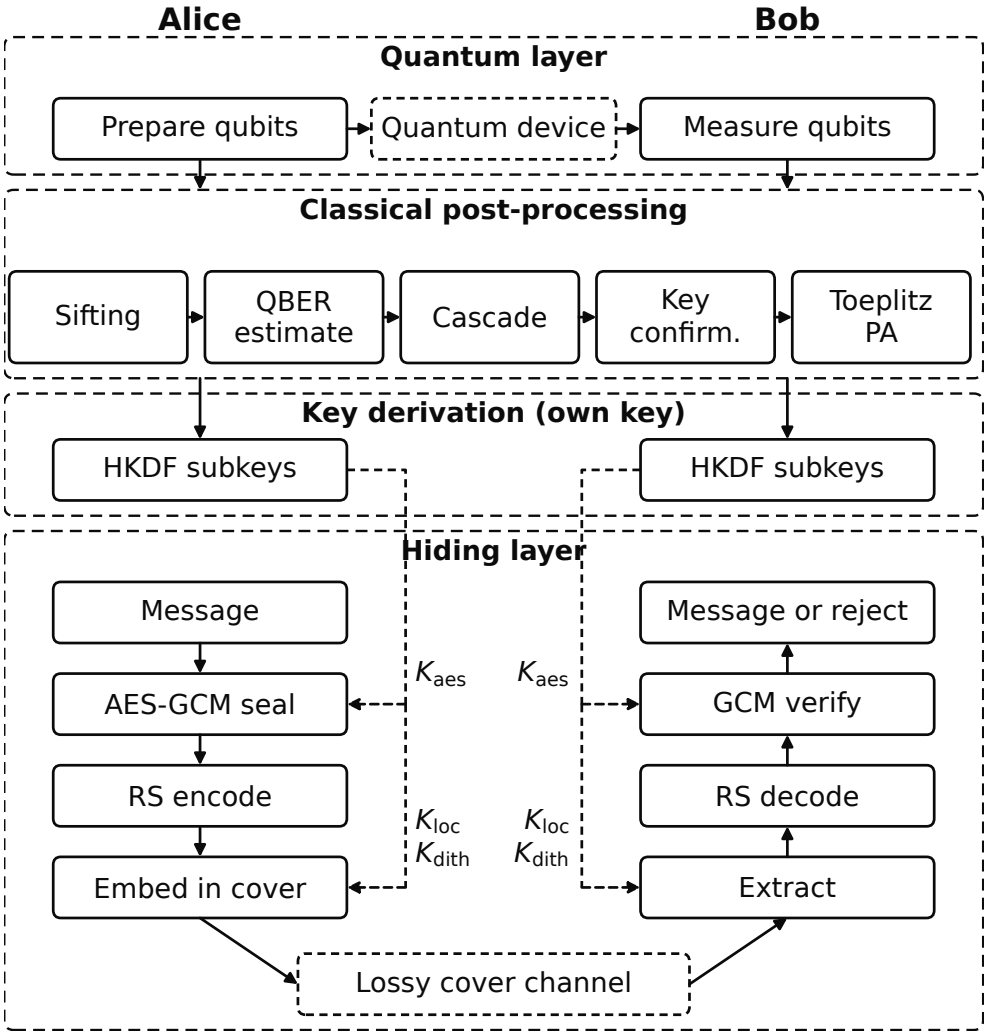


Fig. 1 System architecture. In the quantum layer, Alice's preparation and Bob's measurement run as one circuit on a simulator or an IBM device, on the same qubit; the device contributes noise, not a transmission channel. Classical post-processing (sifting, QBER estimation, Cascade reconciliation, key confirmation and Toeplitz privacy amplification, PA) leaves each party with its own final key, from which it derives HKDF subkeys: K_{aes} for authenticated encryption and K_{loc}, K_{dith} for embedding locations and QIM dither (dashed connectors). Dashed outlines separate the layers. Alice seals, encodes and embeds; after the lossy cover channel, Bob extracts, decodes and either recovers the whole message or rejects it

that neighbouring positions run simultaneously, so crosstalk can correlate their errors. On a cloud device, preparation and measurement run on the same physical qubit within one circuit; the hardware is therefore a source of realistic noise for the key, not a transmission channel.

4.2.2 Randomness

Alice's bits and bases, Bob's bases, the QBER sample and the public seeds are drawn from independent pseudo-random streams spawned from one recorded seed, so every run can be reproduced from its cached record. A deployment-style mode draws all choices from the operating system's cryptographic generator and stores nothing.

4.2.3 Post-processing safeguards

Three choices make the reconciliation step robust at the key lengths available here. First, the Cascade block size (2) is computed from the Clopper–Pearson bound (1) rather than from $\hat{Q}$. With a sample of a few hundred bits, $\hat{Q}$ is frequently an underestimate (including zero on a noisy device), which produces blocks too large to expose errors. Second, the key-confirmation bits are counted as leaked in (6). Third, the disclosed sample bits are excluded from n, so that the hash (3) compresses only bits that never appeared on the public channel. Public seeds for Cascade, confirmation and privacy amplification are fresh for every run.

4.2.4 Eavesdropping

Intercept-resend eavesdropping is simulated classically with ideal-measurement statistics; all observed errors, from any source, are attributed to the eavesdropper.

4.3 Key derivation

As is standard practice, HKDF-SHA256 derives three 256-bit subkeys from the privacy-amplified key, with distinct labels for (i) AES-256-GCM, (ii) embedding-location selection and (iii) the QIM dither. At least 256 secure bits are required. The embedding layer never sees the encryption key.

4.4 Hiding layer

4.4.1 Ordering

The message is length-prefixed, padded to a fixed frame size, sealed with AES-256-GCM, RS-encoded and then embedded. Encryption comes first so that the authentication tag covers the whole frame. Any error pattern that violates (8), including one that causes a miscorrection to a wrong codeword, then causes rejection rather than a wrong message. A transmission succeeds only if the complete message decrypts and its tag verifies.

4.4.2 Embedding arms

Four arms share one key-driven interface:

- *Spatial LSB (reference)*: a representative lossless-channel QKD-keyed scheme and the comparison point for the other arms. Bits replace pixel LSBs after a 32-bit length header, with no RS code. Spatial LSB embedding is used with QKD keys in the literature. [6] Here the pixel order is drawn pseudo-randomly from the location subkey, a standard technique in steganography. [2]

- *DCT-QIM (no ECC)*, where ECC stands for error-correcting code: QIM (7) on the mid-frequency coefficients (zigzag positions 6–20) of the orthonormal 8×8 block DCT, aligned with the JPEG block grid, with no RS code. This arm approximates the closest prior design. [19]
- *DCT-QIM + RS*: the same embedding with an RS code.
- *DWT-QIM + RS*: QIM on the HL and LH detail sub-bands of a level-L Haar wavelet transform, $L \in \{1, 2\}$, with an RS code. The Haar transform is the simplest orthonormal wavelet in the multiresolution framework. [35]

For the QIM arms, the location subkey selects which coefficients carry bits via a keyed permutation, and the dither subkey sets d_0. The selection does not depend on image content, so the receiver finds the same coefficients after distortion.

4.4.3 Minimum step size

After the inverse transform the image is rounded to 8-bit pixels, which perturbs the embedded coefficients. For small Δ, a single modified DCT coefficient changes each pixel by less than half a grey level and rounding undoes it. The embedder therefore re-projects onto the QIM lattices up to four times. The smallest Δ that round-trips exactly with no channel was measured as 16 for DCT-QIM, 4 for DWT-QIM at level 1 and 8 at level 2.

4.5 Channel model

The cover channel applies one of three distortions to the 8-bit grayscale stego image. JPEG recompression uses the standard Independent JPEG Group (IJG) quantization tables at quality 100, 95, 90, 85, 75, 60 or 50. [36] Additive Gaussian noise has a standard deviation of 0.5, 1, 2, 4 or 8 grey levels and is followed by rounding and clipping. Bicubic downscaling by 0.9, 0.75 or 0.5 is followed by upscaling to the original size. A lossless channel, storing the image as portable network graphics (PNG), serves as the control. These are simulated channels and do not reproduce any particular platform, whose encoders may differ or apply several operations in sequence.

5 Experimental setup

5.1 Quantum key generation

5.1.1 Hardware

BB84 was run on ibm_fez (156 qubits), 9 runs in total: $N \in \{256, 1024, 4096\}$ qubits with 3 repeats each, all on one pinned backend. Job identifiers, timestamps, measured bits and seeds are cached, and every key reported here is reproduced from that cache. These are single-day measurements; hardware noise varies between calibration days.

5.1.2 Simulation

To characterize secure key length, 20 BB84 runs per point were simulated for N from 128 to 8,192 on an ideal AerSimulator and with a symmetric readout bit-flip, together

with sweeps over the bit-flip probability and the intercept-resend fraction. A further end-to-end run used the noise model of a fake IBM Heron backend (fake_fez); it is labeled as a noise-model simulation throughout and is not a hardware result.

5.1.3 Keys for the offline experiments

Every trial of the image experiments is keyed by its own simulated BB84 run ($N = 4096$, 3.5% readout flip). Every experiment reports the key-confirmation abort rate of its key source.

5.2 Images

The confirmatory experiments use 300 distinct grayscale images of 512×512 pixels from the 10,000 images of BOSSbase 1.01. [37] They were drawn with a fixed, recorded seed (20261029) before any of them was loaded. The pilot used the 24 images of the Kodak suite, converted to grayscale, with 13 seeds each (312 trials). [38] Trials on the same image are correlated. Detectability is measured on 1,000 further BOSSbase images, disjoint from the confirmatory set (seed 20261030).

5.3 Pilot and confirmatory protocol

The evaluation ran in two stages. The pilot (Kodak) was exploratory, with a 95% success threshold. At JPEG quality 75 it worked only at the largest QIM step in its grid ($\Delta = 32$), and nothing passed at quality 60 or 50, which suggested that the grid ceiling rather than the method limited the result. It also showed that most passing points were limited by capacity on a coarse payload grid.

Before any BOSSbase image was loaded, the confirmatory protocol was recorded. It fixes the following elements:

- the image selection;
- a Δ grid extended to 64 for every arm;
- a payload grid refined near each configuration's capacity;
- a success threshold of 99%;
- the analysis rules.

Each trade-off point is then found in two consecutive stages:

1. *Screening.* Each configuration embeds at full capacity once per image, and the resulting bit-error pattern after recompression determines, exactly for RS bounded-distance decoding (8), which payloads would decode.
2. *Confirmation.* Every reported point is then confirmed by complete end-to-end transmissions on all images, using a best-first search over configurations.

The pre-registered confirmation rule stepped down only within the single best-screened configuration. It was replaced by the best-first search after the results were seen so that every arm is evaluated by the same rule. Rerunning the confirmation on the same screening and seeds changed no reported payload, and both versions are retained.

5.4 Metrics

5.4.1 Authenticated recovery

The primary metric is the fraction of transmissions in which the whole message decrypts and verifies. With k successes in n trials, $\hat{p} = k/n$ and $z = 1.96$, it is reported with the Wilson score interval

$$\frac{\hat{p} + \frac{z^2}{2n} \pm z\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}}, \tag{9}$$

which remains informative at observed rates of 0 and 100%. [39] The raw bit error rate (BER) of the extracted bits before decoding is also reported. A configuration "passes" if its observed recovery rate is at least 99% over the 300 confirmatory images.

5.4.2 Image quality

Distortion between cover X and stego image Y before the channel is measured by the peak signal-to-noise ratio,

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2}{\frac{1}{HW}\sum_{i,j}\left(X_{ij} - Y_{ij}\right)^2}\ \mathrm{dB}, \tag{10}$$

for an $H \times W$ image. The structural similarity index (SSIM) complements it with a measure of local structure. [40] As post-hoc analyses, added after the confirmatory results had been seen, the largest passing payload is also reported subject to a mean PSNR of at least 40 dB (the headline operating point) and 35 dB, together with the 5th-percentile and minimum PSNR over images.

5.4.3 Detectability

Detectability is reported as the area under the receiver operating characteristic curve (AUC) of three detectors:

- the chi-square attack;
- RS steganalysis based on regular and singular pixel groups;
- logistic regression on first-order subtractive pixel adjacency matrix (SPAM) features, augmented with block-DCT coefficient histograms.

The chi-square attack tests whether LSB replacement has equalized the histogram counts of pairs of values. [41] RS steganalysis estimates the LSB embedding rate from how flipping operations change the numbers of regular and singular groups. [42] SPAM features model neighbouring-pixel differences as a Markov chain. [43] The learned detector is evaluated with five-fold cross-validation split by image, so that each cover and its stego version always fall in the same fold and no image appears in both training and test data. Each fold trains on 800 images and tests on 200. A shuffled-label control checks for leakage. These detectors are deliberately simple and far weaker than modern rich-model steganalysis, so measured AUCs are lower bounds on detectability. [44]

5.5 Reproducibility

All experiments are seeded, and the raw per-trial results, hardware caches and seeds are archived together. Every figure and table in this paper regenerates from them without network access. All reported numbers are rounded half up; AUCs are given to three decimals. Two quantities are inherently not reproducible: wall-clock post-processing runtimes, and the unseeded sampling of the local noise-model simulation.

6 Results

6.1 QKD on IBM hardware

Table 1 lists all 9 BB84 runs on ibm_fez. Every run completed without a key-confirmation abort (0/9), and in every run Alice's and Bob's final keys were bit-identical (9/9). The estimated QBER ranged from 0.00% to 0.98%, far below the point at which the asymptotic rate (5) vanishes. [8] At $N = 4096$, Cascade disclosed 142–171 parity bits and the runs yielded 1,110–1,237 secure bits by (6), comfortably above the 256 bits that key derivation requires. At $N = 1024$ the runs yielded 174–187 bits, below that requirement. At $N = 256$ they yielded 0, because the disclosed sample, the reconciliation leakage, the confirmation bits and the security margin together exceed the sifted key. For comparison, the most recent hardware demonstration retained 19 and 23 sifted bits per condition and stopped before reconciliation. [12]

Table 1 BB84 on ibm_fez: all 9 hardware runs, post-processed with the Clopper–Pearson block sizing, Cascade, key confirmation and Toeplitz privacy amplification

N	Run	Sifted	n^1	QBER (%)	L_{EC}	t	ℓ	Keys equal[2]
4096	1	2034	1526	0.98	166	64	1110	yes
4096	2	2090	1568	0.96	171	64	1146	yes
4096	3	2052	1539	0.19	142	64	1237	yes
1024	1	523	393	0.77	59	64	180	yes
1024	2	501	376	0.00	61	64	187	yes
1024	3	504	378	0.79	50	64	174	yes
256	1	127	96	0.00	32	64	0	yes
256	2	123	93	0.00	26	64	0	yes
256	3	127	96	0.00	26	64	0	yes

Source: replay of the cached hardware runs (job identifiers, timestamps, measured bits and seeds in the code repository).

[1]Sifted bits remaining after the QBER sample is disclosed.

[2]Alice's and Bob's final keys compared bit by bit after privacy amplification.

Figure 2 places the hardware runs against 20 simulator runs per point. With an ideal simulator, $N = 1024$ gives 236 ± 11 secure bits (mean $\pm$ standard deviation), and 1/20 runs reach 256 bits. With a 3.5% readout flip, $N = 2048$ gives 270 ± 60 bits (11/20 runs reach 256) and $N = 4096$ gives 650 ± 92 bits (20/20). The hardware points lie between the two simulated curves, consistent with their QBER of at most

0.98%. Of the 720 simulator runs, 608 completed. The other 112 stopped at the QBER threshold, as the protocol requires: 82 in the intercept-resend sweep at eavesdropping fractions of 0.4 and above, 28 in the readout-flip sweep at flip probabilities 0.09–0.12, and 2 with the flip model at $N = 128$ and 256, where the QBER sample is small. In every completed run the keys were bit-identical, and there were 0 key-confirmation aborts. This agrees with the expectation that a correctly sized Cascade leaves residual errors only rarely, and that key confirmation catches them. [27]

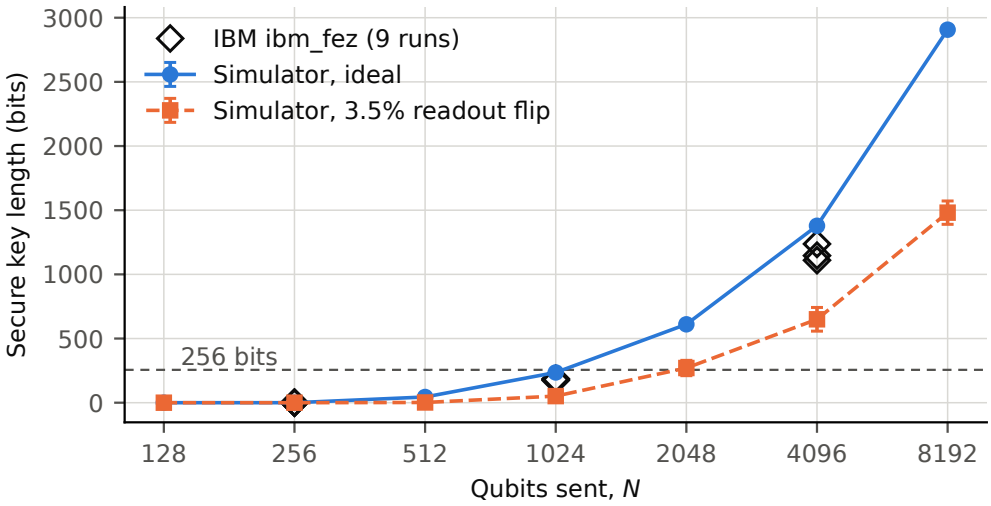


Fig. 2 Secure key length after Cascade, key confirmation and Toeplitz privacy amplification against the number of qubits sent. Simulator points are means with ±1 standard deviation over the completed runs among 20 per point; 2 flip-model runs, at $N = 128$ and 256, aborted at the QBER threshold (circles: ideal; squares: 3.5% readout flip); hollow diamonds are the 9 runs on ibm_fez. The dashed line marks the 256 bits needed for key derivation

In a sweep of the readout flip probability at $N = 4096$, the secure fraction ℓ/n fell from 0.90 at zero QBER to zero at a QBER of about 9.8%, and at 12.0% 15/20 runs aborted at the threshold. The zero crossing lies below the 11% of (5) because (6) also subtracts the measured Cascade leakage, the confirmation bits and the margin. Simulated intercept-resend eavesdropping produced points on the same curve as readout noise: the protocol sees only the QBER, not its cause. Noise from decoherence and from an eavesdropper is likewise indistinguishable to the protocol in recent analyses of BB84. [10] Privacy amplification is the slowest step in the pure-Python implementation, taking 0.15–0.88 s per run at $N = 4096$ and up to 3.2 s at $N = 8192$. The hardware figures are single-day measurements on one backend; device noise varies between calibration days.

6.2 End-to-end transmission with hardware-derived keys

Table 2 shows complete transmissions keyed by BB84 keys. The first $N = 4096$ hardware run (job `dbgu091q260s73bu26vg`, QBER 0.98%, 1,110 secure bits) was replayed from the cache. Alice and Bob derived their subkeys from their own keys with HKDF. [32] A message was then embedded in a Kodak image with the setting that Table 3 supports at its most lossy quality: DCT-QIM + RS with $\Delta = 64$ and $n_{\mathrm{sym}} = 128$. Bob decrypted and verified the message both over a lossless channel and after JPEG recompression at quality 50, the harshest compression tested. The extracted bits had a raw error rate of 1.3% before Reed–Solomon decoding, and the stego image had a PSNR of 45.8 dB. The same transmission keyed by the noise-model simulation (fake_fez, QBER 0.20%, 1,297 secure bits) also verified, with a raw error rate of 1.9%. In both cases the GCM tag verified, which by construction of the mode means that the recovered frame is bit-identical to the transmitted one. [7]

Table 2 End-to-end transmissions keyed by BB84 with $N = 4096$: Bob extracts, decodes and decrypts with his own reconciled key. Setting from the 40 dB table at its most lossy JPEG quality: DCT-QIM + RS, $\Delta = 64$, $n_{\mathrm{sym}} = 128$

Key source	QBER (%)	ℓ	Lossless	JPEG Q50	Raw BER (%)[1]
Hardware (ibm_fez)	0.98	1,110	verified	verified	1.3
Noise model (fake_fez)	0.20	1,297	verified	verified	1.9

The noise-model row is a simulation with a fake-backend noise model, not a hardware result.

[1] Bit error rate of the extracted bits after JPEG, before Reed–Solomon decoding.

These are single demonstrations, not statistics. The statistics in the rest of this section use keys from simulated BB84 runs (Sect. 5.1). This is sufficient by construction: the hiding layer sees only the HKDF subkeys derived from the final key, so its behavior does not depend on whether that key came from hardware or from simulation, provided Alice's and Bob's keys agree. The hardware-keyed transmission above demonstrates this directly for one run; it is not a statistical comparison of hardware and simulated keys.

6.3 The LSB reference under JPEG recompression

Figure 3 shows the motivating failure on the 300 confirmatory images. The LSB reference scheme delivered 300/300 messages over a lossless channel and 0/300 at every JPEG quality tested, including quality 100; the upper Wilson bound (9) is 1.3%. The raw bit error rate was 0.097 at quality 100 and 0.488–0.500 at quality 95 and below, which is chance level. The pilot agrees: on Kodak, the LSB reference delivered 624/624 messages losslessly and 0/2496 at qualities 95, 90, 75, 50, with raw error rates of 0.497–0.501. JPEG requantizes the DCT coefficients of every 8×8 block, which leaves the pixel LSBs essentially random. [36] The 32-bit length header is lost as well,

so the receiver cannot even determine how many bits to read. The fragility of spatial LSB embedding to any lossy processing is well documented in surveys of image steganography. [2]

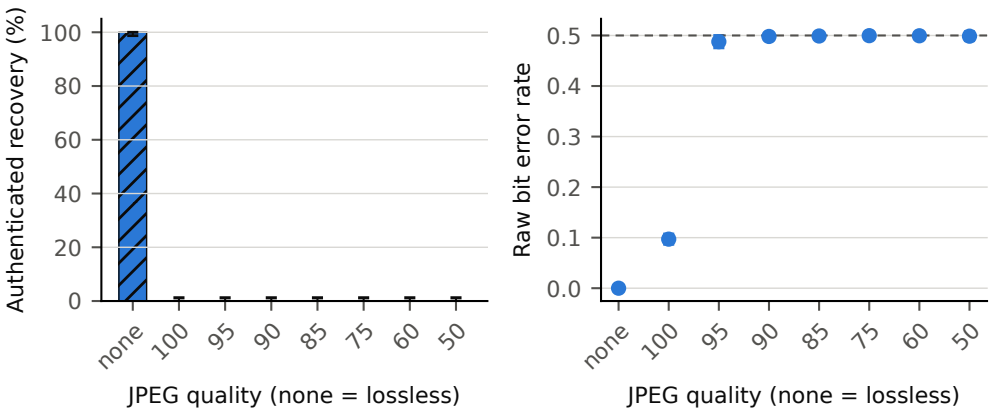


Fig. 3 Spatial LSB (reference) on 300 BOSSbase images. Left: authenticated recovery with Wilson 95% intervals. Right: raw bit error rate of the extracted bits (mean ±1 standard deviation); the dashed line marks chance level

6.4 Robustness at the pre-registered defaults

Figures 4 and 5 use the pre-registered defaults ($\Delta = 16$, $n_{\text{sym}} = 32$, DWT level 1, message payload 0.02 bits per pixel, bpp).

6.4.1 JPEG

DCT-QIM + RS delivered 300/300 messages at quality 95 [98.7%, 100.0%] and 2/300 at quality 90. DWT-QIM + RS delivered 300/300 at quality 100 and 0/300 at quality 95, where its raw error rate was already 0.078. Recovery is close to all-or-nothing: a configuration delivers essentially every message or almost none, and the transition happens between adjacent quality levels. This matches the QIM decision rule of (7), under which a coefficient decodes correctly as long as the channel moves it by less than $\Delta/4$. [17]

6.4.2 Error correction is essential

Without Reed–Solomon coding, DCT-QIM delivered 0/300 messages at quality 95 even though its raw error rate was only 0.0029 (and 0.0012 at quality 100). A handful of bit errors anywhere in the frame is enough for the authentication tag to reject the message. With $n_{\text{sym}} = 32$, the same errors fall well within the correction limit of (8), which explains the difference between the coded and uncoded DCT arms. [34]

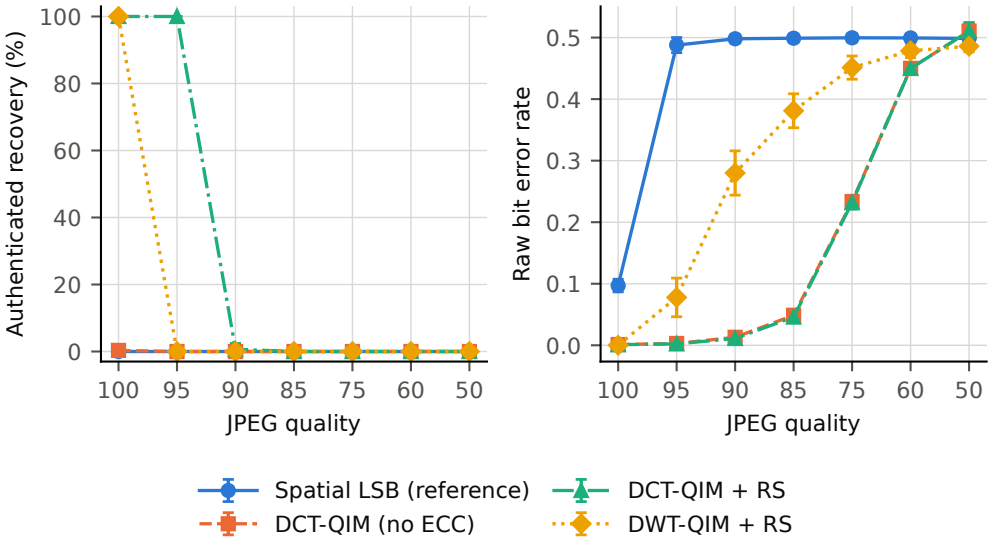


Fig. 4 Robustness to JPEG recompression at the pre-registered defaults on 300 BOSSbase images; qualities are evenly spaced along the horizontal axis. Left: authenticated recovery with Wilson 95% intervals. Right: raw bit error rate before decoding (mean ±1 standard deviation). Each arm has its own marker and line style; where several arms deliver nothing, their markers overlap at 0%

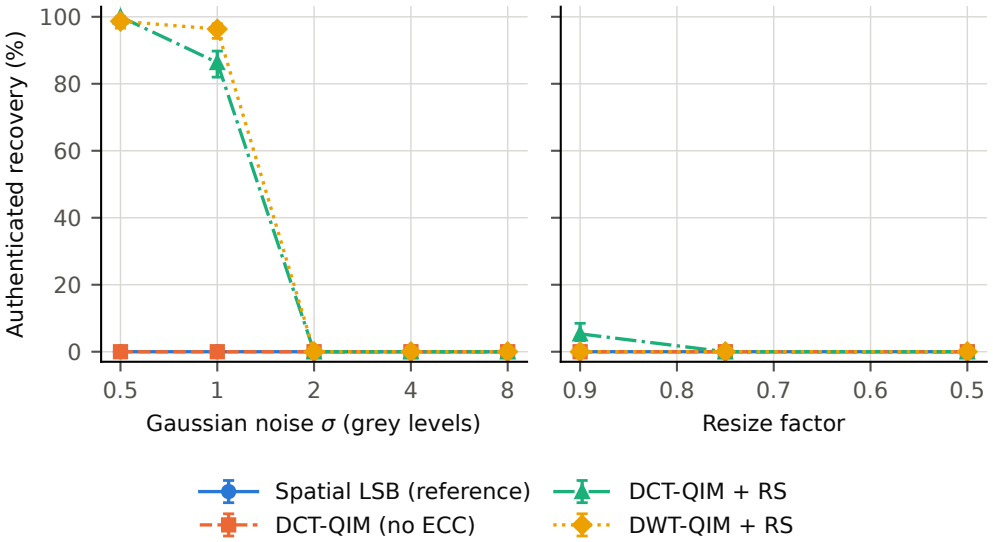


Fig. 5 Authenticated recovery under additive Gaussian noise (left) and under downscaling followed by upscaling to the original size (right), at the pre-registered defaults, with Wilson 95% intervals; where several arms deliver nothing, their markers overlap at 0%

6.4.3 Noise and resizing

Under Gaussian noise with $\sigma = 1$ grey level, DWT-QIM + RS delivered 289/300 [93.6%, 97.9%] and DCT-QIM + RS 259/300 [82.0%, 89.8%]; at $\sigma \geq 2$ no arm delivered any message. Resizing defeated every arm: the best case was 16/300 [3.3%, 8.5%], for DCT-QIM + RS at scale 0.9. Robust steganography that targets JPEG typically assumes that the block grid is preserved, which resampling destroys. [15]

6.4.4 Step size and parity at quality 75

Over the full $\Delta \times n_{\text{sym}}$ grid at quality 75 and 0.02 bpp, DCT-QIM + RS reached the 99% criterion only for $\Delta \geq 32$, and at $\Delta = 32$ only with $n_{\text{sym}} \geq 96$; larger steps needed fewer parity bytes (Fig. 10 in Appendix A). DWT-QIM + RS at level 1 passed in 0 grid cells.

6.5 Capacity, image quality and robustness

6.5.1 Headline (post-hoc): PSNR of at least 40 dB

Table 3 and Fig. 6 give the largest payload that each arm delivers with at least 99% authenticated recovery while keeping the mean PSNR (10) at or above 40 dB. This floor was chosen after the confirmatory results had been seen, so the table is a post-hoc analysis; it is the headline because it describes operating points without visible distortion. The table reports the mean, 5th-percentile and minimum PSNR over the 300 images, and every single image in every row is above 40 dB (the lowest is 40.1 dB). Lower Wilson bounds for the passing rows lie between 97.1% and 98.7%.

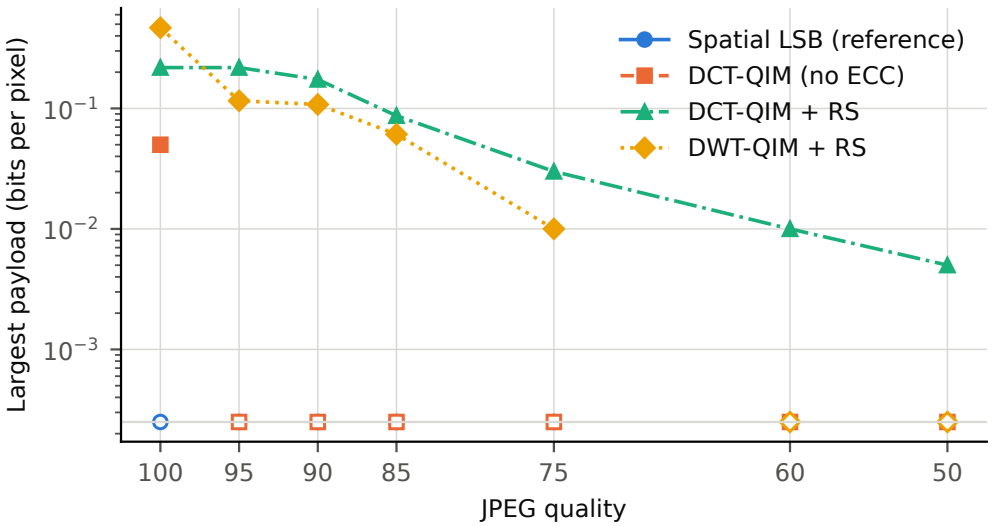


Fig. 6 Post-hoc headline: largest payload with at least 99% authenticated recovery and mean PSNR of at least 40 dB against JPEG quality, confirmed end to end on 300 images. Hollow markers on the bottom line mean that no payload passes

Table 3 Post-hoc analysis (headline): largest message payload with at least 99% authenticated recovery and mean PSNR of at least 40 dB, confirmed end to end on all 300 confirmatory images. Rows are shown only where a payload passes

Arm	Q	Δ	n_{sym}	L	bpp	Recovery (%)[2]	PSNR (dB)[3]
DCT-QIM + RS	100	8	16	–	0.218[1]	100.0 (98.7)	46.9/46.8/46.7
	95	12	16	–	0.218[1]	100.0 (98.7)	43.5/43.4/43.1
	90	16	64	–	0.174[1]	100.0 (98.7)	41.1/41.0/40.7
	85	24	64	–	0.087	100.0 (98.7)	40.5/40.5/40.1
	75	32	128	–	0.030	100.0 (98.7)	40.8/40.8/40.6
	60	48	128	–	0.010	99.3 (97.6)	41.7/41.6/41.5
	50	64	128	–	0.005	99.7 (98.1)	41.4/41.2/41.2
DWT-QIM + RS	100	8	16	1	0.467[1]	100.0 (98.7)	43.7/43.7/43.6
	95	24	16	2	0.116[1]	99.7 (98.1)	40.3/40.3/40.2
	90	24	32	2	0.108[1]	99.0 (97.1)	40.3/40.3/40.2
	85	24	128	2	0.061[1]	99.7 (98.1)	40.3/40.3/40.2
	75	48	128	2	0.010	99.0 (97.1)	41.7/41.6/41.5
DCT-QIM (no ECC)	100	24	–	–	0.050	99.3 (97.6)	44.1/44.1/43.9

The PSNR floor was chosen after the confirmatory results had been seen. Spatial LSB (reference): no payload passes at any quality. DWT-QIM + RS: none at Q60 or Q50. DCT-QIM (no ECC): none below Q100.

[1]Capacity-limited: the next larger payload on the configuration's grid does not fit.

[2]Observed rate over 300 images; Wilson 95% lower bound in parentheses.

[3]Mean / 5th percentile / minimum over the 300 stego images.

DCT-QIM + RS is the only arm that delivers messages at every quality tested. Its payload falls from 0.218 bpp (7,152 bytes per 512 × 512 image) at quality 95 to 0.030 bpp at 75, 0.010 bpp at 60 and 0.005 bpp (164 bytes) at 50. DWT-QIM + RS carries the most at quality 100 (0.467 bpp) but falls behind from quality 95 downwards, reaching 0.010 bpp at 75 and nothing at 60 or 50. The uncoded DCT arm passes only at quality 100, and the LSB reference nowhere. In 7/13 passing rows the next larger payload would not fit at all, so capacity rather than bit errors sets the limit. Dedicated robust JPEG steganography with specialized codes reports error-free extraction at higher payloads, so these figures describe simple coded QIM rather than the state of the art. [16]

6.5.2 Relaxed and unconstrained

Table 4 repeats the analysis with a 35 dB floor (also post-hoc) and without a floor (pre-registered). The 35 dB floor applies to the mean; in 4 passing configurations some images fall slightly below it, the lowest at 34.4 dB. Without a floor, DCT-QIM + RS reaches 0.218 bpp at every quality down to 75 and 0.115 bpp at 50, but at quality 75 and below this requires $\Delta = 64$ and costs a mean PSNR of about 29.1 dB, which is visible. DWT-QIM + RS reaches 0.467 bpp at quality 100 and 0.077 bpp at 75. Here 12/13 passing points are capacity-limited. Figure 7 shows the trade-off between payload and PSNR at quality 75 across all configurations; its points are screening estimates, and the reported operating points are those confirmed in the tables.

Table 4 Largest payload in bits per pixel (mean PSNR in dB) with at least 99% authenticated recovery: with a 35 dB mean-PSNR floor (post-hoc) and without a floor (pre-registered)

	PSNR $\geq$ 35 dB[1]		No PSNR floor[2]	
Q	DCT-QIM + RS	DWT-QIM + RS	DCT-QIM + RS	DWT-QIM + RS
100	0.218 (46.9)	0.467 (43.7)	0.218 (46.9)	0.467 (43.7)
95	0.218 (43.5)	0.316 (35.1)	0.218 (43.5)	0.436 (28.4)
90	0.218 (37.6)	0.108 (40.3)	0.218 (37.6)	0.310 (26.0)
85	0.218 (35.1)	0.092 (37.9)	0.218 (35.1)	0.092 (37.9)
75	0.115 (35.1)	0.065 (35.0)	0.218 (29.1)	0.077 (34.4)
60	0.030 (37.4)	–	0.174 (29.1)	–
50	0.020 (36.3)	–	0.115 (29.1)	–

–: no payload passes. Spatial LSB (reference) passes nowhere in either analysis.

[1]Post-hoc; the floor applies to the mean. The lowest single-image PSNR in this column pair is 34.4 dB.

[2]Pre-registered analysis, re-run with the best-first confirmation rule for every arm; no payload changed.

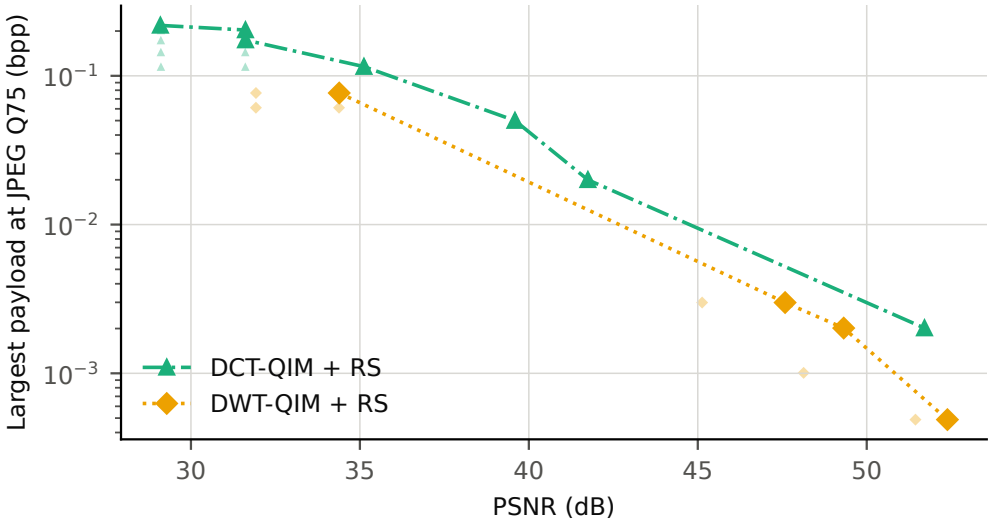


Fig. 7 Payload against PSNR at JPEG quality 75 for every configuration of the two coded QIM arms (small markers, screening estimates), with each arm's Pareto front (line)

6.5.3 DCT against DWT

The two transforms divide the channel conditions between them. DCT-QIM is more robust to JPEG and DWT-QIM to additive noise (Sect. 6.4), and at level 1 DWT-QIM has about twice the embedding capacity of DCT-QIM (two detail coefficients per 2×2 pixels against 15 coefficients per 8×8 block), which is why it leads at quality 100. A plausible reason for the JPEG advantage, which has not been tested here, is that the DCT arm embeds in the same 8×8 orthonormal DCT, on the same block grid, that

JPEG quantizes, so a QIM step larger than the JPEG quantization step of a coefficient may survive recompression directly. A Haar coefficient, by contrast, mixes several DCT frequencies, because the wavelet and block-DCT bases do not coincide. [35]

6.6 Image quality and detectability

6.6.1 Image quality

Distortion is set mainly by the QIM step and the payload, not by the transform. At 0.02 bpp, the PSNR of DCT-QIM + RS was 50.4, 44.8 and 38.9 dB for $\Delta = 16$, 32 and 64, about 5.8 dB per doubling of Δ, close to the 6 dB per doubling that (10) predicts when the mean squared error grows with Δ^2. DWT-QIM + RS gave 50.8 dB at $\Delta = 16$. At $\Delta = 64$ the SSIM of DCT-QIM + RS was 0.934, which indicates that local structure is largely preserved even at the largest step. [40] The LSB reference, which changes only least significant bits, reached 67.9 dB at the same payload and 61.1 dB at 0.1 bpp. Figure 8 shows one Kodak image and the amplified embedding changes of each arm. For the DCT arms below the minimum step of Sect. 4.4, rounding erases much of the embedding, so their high PSNR does not indicate imperceptibility.

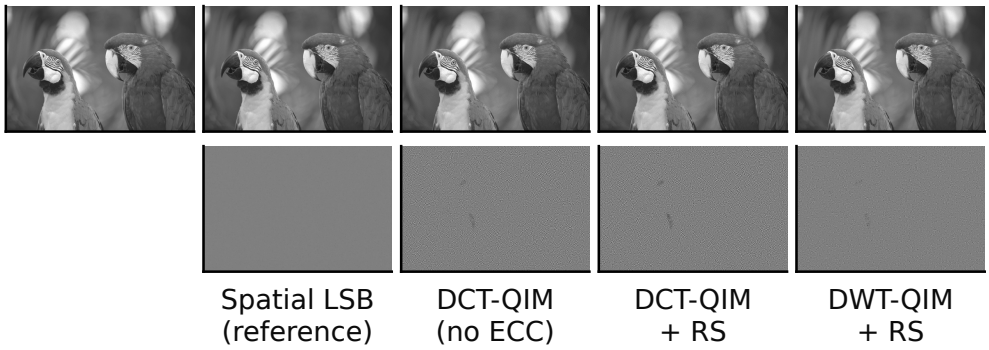


Fig. 8 Cover image (top left) and, per arm, the stego image (top row) and the difference from the cover amplified 16 times around mid-grey (bottom row), for a 0.05 bpp message and $\Delta = 32$. The cover is image kodim23 of the Kodak suite. [38]

6.6.2 Detectability

Figure 9 and Table 5 report detection AUC on 1,000 BOSSbase images that are disjoint from the confirmatory set. The learned detector was trained and tested with a split by image: in each of five folds it trained on 800 images and tested on 200, with every cover and its stego version in the same fold. Its shuffled-label control stayed between 0.484 and 0.527.

The robustness of the QIM arms comes at a clear cost in detectability. The logistic-regression detector identified every QIM arm with an AUC of 0.797–0.945 at the smallest payload tested (0.005 bpp), and of at least 0.985 from 0.05 bpp upwards. The

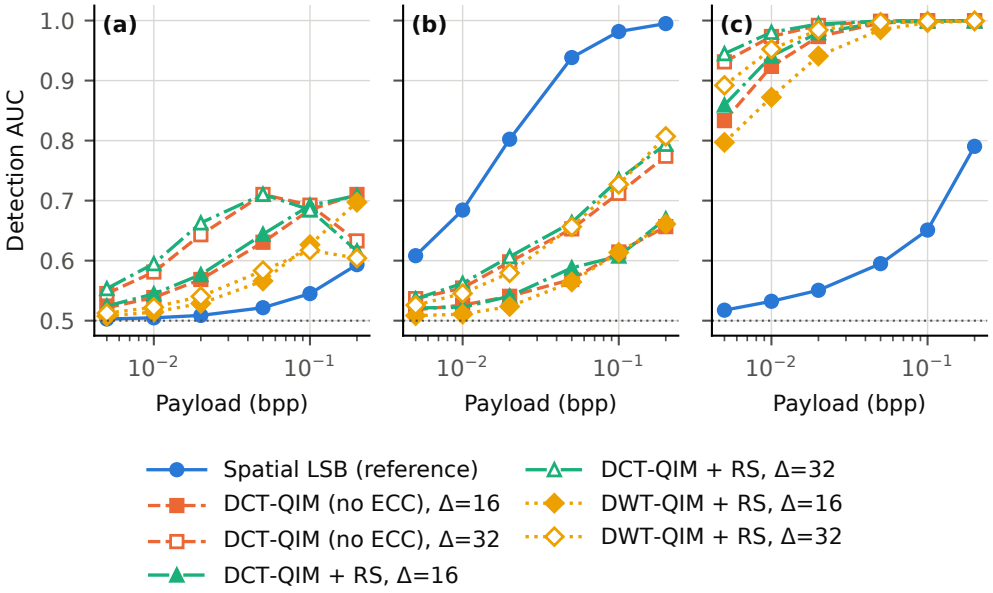


Fig. 9 Detection AUC against message payload on 1,000 BOSSbase images disjoint from the confirmatory set: (a) chi-square attack, (b) RS analysis, (c) logistic regression on SPAM and block-DCT histogram features with a five-fold split by image. Filled markers: $\Delta = 16$; hollow markers: $\Delta = 32$; the dotted line marks chance level

Table 5 Detection AUC of the logistic-regression detector (SPAM and block-DCT histogram features) on 1,000 BOSSbase images disjoint from the confirmatory set, five-fold cross-validation split by image

	Message payload (bpp)			
Arm	0.005	0.02	0.05	0.2
Spatial LSB (reference)	0.517	0.551	0.595	0.790
DCT-QIM (no ECC), $\Delta = 16$	0.833	0.973	0.996	1.000
DCT-QIM (no ECC), $\Delta = 32$	0.931	0.992	0.999	0.999
DCT-QIM + RS, $\Delta = 16$	0.859	0.980	0.997	1.000
DCT-QIM + RS, $\Delta = 32$	0.945	0.994	0.999	0.999
DWT-QIM + RS, $\Delta = 16$	0.797	0.941	0.985	0.999
DWT-QIM + RS, $\Delta = 32$	0.892	0.984	0.997	1.000

Each fold trains on 800 images and tests on 200 images; covers and their stegos share a fold. Shuffled-label control: AUC 0.484–0.527. QIM arms use $n_{\mathrm{sym}} = 32$ for the coded variants.

same detector reached only 0.517, 0.595 and 0.790 on the LSB reference at 0.005, 0.05 and 0.2 bpp. The LSB reference is not undetectable either: RS analysis detected it with an AUC of 0.608 [0.600, 0.617] at 0.005 bpp, 0.938 at 0.05 bpp and 0.995 at 0.2 bpp. This is consistent with the design of RS analysis, which targets exactly this kind of LSB replacement. [42] On the QIM arms RS analysis gave 0.508–0.807 and the

chi-square attack 0.508–0.711. Because both detectors target LSB replacement, these values say little about the security of QIM. On the LSB reference the chi-square attack gave 0.503–0.593, as expected for a message scattered at low rates, which the global attack was not designed to find. [41] The learned detector uses features that include block-DCT coefficient histograms, which observe the DCT band directly, alongside SPAM features of pixel differences. [43] It is nonetheless far weaker than modern rich-model steganalysis, so its AUCs are lower bounds on how detectable the robust arms are. [44]

6.7 Ablations

6.7.1 Uncoded DCT-QIM

The uncoded arm approximates the closest prior design, which reports bit accuracy rather than whole-message recovery. [19] It delivered messages only at quality 100: 0.210 bpp without a PSNR floor and 0.050 bpp with the 40 dB floor. At qualities 95 and 90 the best-first confirmation tried 19 and 6 candidate points and none passed, although its bit accuracy at quality 95 exceeded 99% (raw error rate 0.0029). High bit accuracy and no delivered messages are compatible, which is why authenticated recovery is the reported metric.

6.7.2 Erasure decoding

Marking unreliable bytes as erasures, using the QIM decision margin with a policy fixed before the run, did not help. By (8), an erasure costs half as much correction capacity as an error, so erasures help only if they are placed on the bytes that are actually wrong. Over 24,000 paired transmissions in 80 configurations of the coded arms at JPEG qualities 95 to 75, erasures rescued 51 messages and lost 78, and the per-configuration change in recovery ranged from -3.0 to +1.0 percentage points. The likely interpretation, which was not measured, is that under JPEG a corrupted coefficient tends to land confidently on the wrong lattice point rather than near the decision boundary, so that a small margin is a poor indicator of a bit error.

7 Discussion

7.1 Two transforms, two channels

No single variant is best everywhere. DCT-QIM + RS is the arm to use when JPEG recompression is expected, and the only one that delivers messages at quality 60 and 50 (Table 3). DWT-QIM + RS tolerates additive noise better and carries more when compression is mild. A deployment that does not know its channel would therefore have to choose conservatively or estimate the channel first.

7.2 Why bit accuracy is the wrong metric

Watermarking studies typically report bit accuracy, and a bit accuracy above 99% sounds like success. For an authenticated payload it is not: the uncoded DCT arm

reached such accuracy at quality 95 and delivered no message at all (Sect. 6.7). What matters for an encrypted, authenticated message is whether every bit of the frame survives error correction. This is why recovery is all-or-nothing, and why whole-message recovery is reported with confidence intervals (9) rather than mean bit accuracy.

7.3 The robustness–detectability trade-off

The transform-domain arms survive JPEG because they write a strong, structured signal into coefficients that recompression largely preserves. The same structure is what a detector sees. Even a deliberately simple detector separates them from covers far more easily than it separates the LSB reference (Sect. 6.6). The system studied here therefore provides robust, authenticated delivery of QKD-keyed messages. It does not conceal the presence of a message from a capable steganalyst and should not be described as doing so. Adaptive embedding that resists both JPEG compression and detection is one route by which the hiding layer could improve. [18] Selecting the embedding domain to match the channel is another. [15]

7.4 Security analysis and limitations

7.4.1 Key distribution

- The secure key length (6) is an asymptotic, teaching-level estimate, not a finite-key-rigorous bound. The QBER is a point estimate from a 25% sample, and a finite-key treatment would replace both with statistically bounded quantities. [31]
- All QBER is attributed to the eavesdropper, including gate, readout and decoherence errors; BB84 cannot separate noise from eavesdropping. The 11% abort threshold marks where the key rate vanishes; it is not a test that distinguishes an eavesdropper from noise.
- On IBM hardware, Alice's preparation and Bob's measurement run on the same qubit within one circuit. This is not a transmission channel: there is no distance, no loss and no real interception.
- On cloud quantum hardware the provider can see Alice's preparation gates, so the keys derived in these experiments are not secret from the platform. The hardware runs characterize noise and pipeline behavior, not deployable key secrecy.
- Alice's and Bob's random choices come from a seeded pseudo-random generator for reproducibility, not from a quantum random number generator as a real deployment would use. A deployment-style mode that draws all choices from the operating system's cryptographic generator exists but was not used for the reported runs.
- The seeds of all runs are recorded, so every experimental key can be reconstructed. These keys demonstrate the pipeline and must never protect real data.
- Eavesdropping is simulated as intercept-resend only, and classically, even when Bob's measurements come from hardware.
- Up to 100 BB84 positions run in parallel on neighboring qubits of one circuit; crosstalk can correlate their errors and is not modeled.
- The hardware results come from one backend on one day (9 runs). The noise-model run is a calibration snapshot, not the device.

- The message is protected by AES-256 under the QKD-derived key, so its confidentiality is computational, as for any QKD-keyed symmetric cipher; it is not information-theoretic.
- Cascade reconciliation, Toeplitz privacy amplification, HKDF and the 12-byte GCM nonce are standard components, integrated here; they are not contributions of this work.

7.4.2 Hiding layer and channel

- Transform-domain embedding improves robustness, not undetectability. It is more detectable than the LSB reference (Sect. 6.6), and the embedding is not claimed to match state-of-the-art robust steganography.
- The channel is simulated, with one JPEG encoder and the standard quantization tables, additive Gaussian noise and bicubic resizing. Real platforms may use other encoders, compress twice, resize or filter; no claim is made about any particular platform.
- Resizing defeats every arm (Sect. 6.4). Geometric attacks such as cropping, shifting and rotation were not tested and would break the block-grid alignment on which the DCT arm relies.
- Strong compression costs quality and capacity. Above 40 dB only DCT-QIM + RS delivers messages at quality 60 and 50, and only small ones; larger payloads at those qualities need $\Delta = 64$ and visible distortion. The step size was not extended beyond 64, because a larger step only lowers PSNR further below both floors.
- Steps below a minimum value do not survive rounding to 8-bit pixels (Sect. 4.4).
- Soft-decision erasure decoding with the fixed policy tested did not help (Sect. 6.7).

7.4.3 Statistics and procedure

- The pilot (Kodak, 24 images $\times$ 13 seeds) is exploratory; its trials on the same image are correlated.
- The 99% criterion is an observed rate on 300 images; the lower Wilson bounds of passing configurations are 97.1–98.7%. It is not a guarantee.
- Payload maxima depend on the payload grid: they are conservative between grid points, and most are capacity-limited.
- The stage-2 confirmation rule was changed after the results had been seen: the best-first search over all configurations replaced stepping down within the single best-screened configuration. It was applied to all arms, on the same screening and seeds, and changed no reported payload.
- The PSNR floors, and therefore Tables 3 and 4 (35 dB columns) and Fig. 6, are post-hoc. The floors apply to the mean PSNR over images.
- Screening estimates the effect of a smaller payload from a full-capacity error pattern; it selects candidates only. Every reported operating point is confirmed end to end, but Fig. 7 shows screening estimates.
- The detectors are deliberately simple. The chi-square attack also fires on some clean, textured covers, RS analysis breaks down near full embedding, and the learned detector uses hand-crafted features, not a modern steganalyzer.

- Post-processing runtimes are wall-clock measurements, and the local noise-model simulation is not seeded; both vary between runs. All other numbers regenerate exactly from the archived results.

7.5 Future work

Several directions follow from these results and limitations.

7.5.1 Finite-key security and reconciliation

The secure key length (6) should be replaced by a finite-key bound that accounts for the statistical uncertainty of the QBER sample and for a non-zero failure probability of each post-processing step. [31] At the key lengths available on current devices, such a bound will shorten the final key, so the qubit budget per message needs to be re-derived under it. Reconciliation itself can also be made more efficient. Codes and protocols that disclose fewer parity bits than Cascade would raise the secure fraction at a given QBER. [14]

7.5.2 An authenticated, physically separated key exchange

The classical discussion between Alice and Bob is assumed to be authenticated but is not authenticated in the present implementation. Wegman–Carter authentication with a short pre-shared key, built from the same universal hash families used for key confirmation, would close this gap. [28] Keys should also come from a physically separated QKD link with quantum random number generation, so that the hardware layer contributes secrecy and not only realistic noise. [9]

7.5.3 Hardware variability

The hardware results come from one backend on one day. Repeated runs across calibration days, backends and qubit layouts would show how much the QBER and the secure key length vary in practice, and whether crosstalk between the parallel positions of a wide circuit matters. Analyses of how decoherence affects BB84 security provide a model against which such measurements could be compared. [10]

7.5.4 Lower detectability at the same robustness

The main limitation of the hiding layer is its detectability. Adaptive embedding that concentrates changes in textured regions, combined with channel-aware coding, could retain the robustness of the coded DCT arm while reducing the detector's AUC. [18] Steganographic polar codes are a second candidate, because they achieve error-free extraction after recompression with structured, low-redundancy coding. [16] Any such design should be evaluated against modern rich-model and learned steganalyzers rather than the simple detectors used here. [44]

7.5.5 Real platform channels

The simulated channel should be extended to the operations that real platforms apply: double compression with different quantization tables, resizing, cropping and filtering. Robustness to resizing and other geometric transformations requires embedding domains that do not depend on a fixed block grid. [15] Colour images, which most platforms carry, would add capacity in the chrominance channels but are also recompressed with coarser quantization.

7.5.6 Faster post-processing and quantum image representations

Privacy amplification is the slowest step of the pure-Python implementation, and a fast-Fourier-transform implementation of the Toeplitz product would make longer keys practical. Finally, as quantum hardware scales, hiding data directly in quantum image representations such as NEQR may become feasible on devices, and a combined system could compare classical and quantum hiding layers under the same key distribution. [24]

8 Conclusion

This work presents a pipeline in which BB84 keys obtained on IBM quantum hardware, after Cascade reconciliation, key confirmation and Toeplitz privacy amplification, drive authenticated information hiding. At $N = 4096$, all hardware runs produced bit-identical keys long enough for key derivation (Table 1). Placing Reed–Solomon coding between authenticated encryption and key-driven transform-domain embedding makes the hiding layer work over lossy image channels. The LSB reference scheme delivers no message after JPEG recompression, whereas coded DCT-QIM delivers whole, verified messages at every JPEG quality tested while staying above 40 dB PSNR, with capacity shrinking as compression grows (Table 3). A message keyed by a hardware-derived key decrypted and verified after JPEG quality 50 (Table 2). This robustness has a measured price: a simple learned detector identifies the robust variants far more easily than the LSB reference (Table 5).

Acknowledgements. The authors thank the Department of Computer Science and Engineering, COEP Technological University, Pune, for providing the computing resources, infrastructure and academic environment in which this work was carried out. The authors also acknowledge the use of IBM Quantum services for this work.

Declarations

- **Funding:** This work was supported by the Maharashtra Government Quantum Mission (Research Funding: Grant No. 1381878/HTED-11011(11)/13/2025-MHT-(TE-2)).
- **Competing interests:** Not applicable.
- **Ethics approval and consent to participate:** Not applicable.
- **Consent for publication:** Not applicable.
- **Data availability:** Not applicable.

- **Materials availability:** Not applicable.
- **Code availability:** Not applicable.
- **Author contribution:** A.S., A.U., H.S. and K.A. contributed equally to conceptualization, methodology, software, validation, formal analysis, investigation, data curation, visualization and writing of the original draft. M.N. mentored and guided the work, contributed to conceptualization and supervision, and reviewed and edited the manuscript. S.B. provided overall guidance, supervision and project administration, and reviewed and edited the manuscript. All authors read and approved the final manuscript.

Appendix A Step size and parity at JPEG quality 75

Figure 10 shows authenticated recovery for every combination of QIM step Δ and parity bytes per codeword n_{sym} on the 300 confirmatory images, after JPEG recompression at quality 75 with a payload of 0.02 bpp. These are the cells summarized in Sect. 6.4.

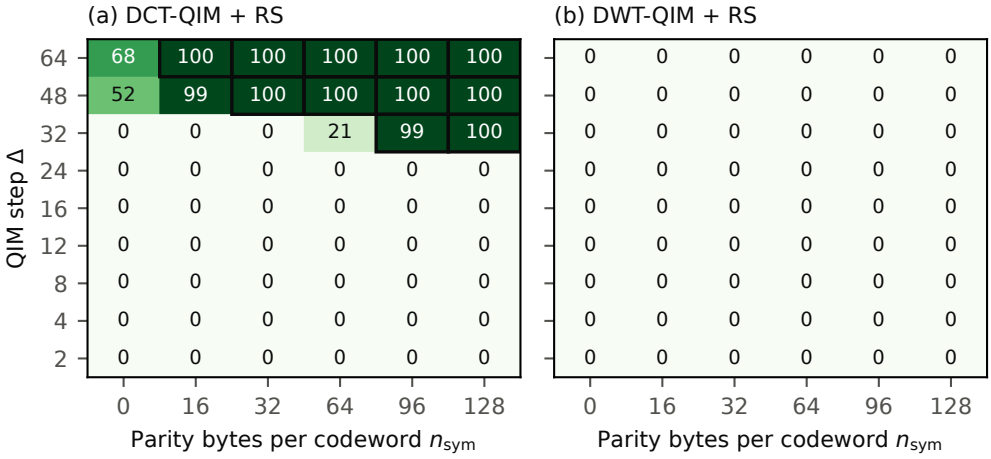


Fig. 10 Authenticated recovery (%) over the $\Delta \times n_{\text{sym}}$ grid at JPEG quality 75 and 0.02 bpp on 300 BOSSbase images: (a) DCT-QIM + RS, (b) DWT-QIM + RS at level 1. Outlined cells meet the 99% criterion. Values are rounded to whole percentages

References

[1] Bennett, C.H., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. Theor. Comput. Sci. **560**, 7–11 (2014) https://doi.org/10.1016/j.tcs.2014.05.025

[2] Cheddad, A., Condell, J., Curran, K., Mc Kevitt, P.: Digital image steganography: Survey and analysis of current methods. Signal Process. **90**(3), 727–752 (2010) https://doi.org/10.1016/j.sigpro.2009.08.010

[3] Sykot, A., Azad, M.S., Tanha, W.R., Morshed, B.M.M., Shubha, S.E.U., Mahdy, M.R.C.: Multi-layered security system: Integrating quantum key distribution with classical cryptography to enhance steganographic security. Alex. Eng. J. **121**, 167–182 (2025) https://doi.org/10.1016/j.aej.2025.02.056

[4] Rezaeishad, D.S., Bahramgiri, H.: Dual-layered quantum-secure concealing: Steganography over quantum key distribution. ISC Int. J. Inf. Secur. **18**(3), 287–296 (2026) https://doi.org/10.22042/isecure.2026.240529

[5] Mogos, G., Lam, H.-P.: Quantum steganography for secure data transmission. Cybersecurity **9**(1), 118 (2026) https://doi.org/10.1186/s42400-025-00529-4

[6] Arumugam, A., Eswaran, B., Ramesh, L., Nallusamy, C.: Quantum-integrated steganography for secure communication using QKD and LSB techniques. In: 2025 International Conference on Electronics and Renewable Systems (ICEARS), pp. 977–983. IEEE, ??? (2025). https://doi.org/10.1109/ICEARS64219.2025.10940779

[7] Dworkin, M.J.: Recommendation for block cipher modes of operation: Galois/Counter Mode (GCM) and GMAC. NIST Special Publication 800-38D, National Institute of Standards and Technology (2007). https://doi.org/10.6028/NIST.SP.800-38D

[8] Shor, P.W., Preskill, J.: Simple proof of security of the BB84 quantum key distribution protocol. Phys. Rev. Lett. **85**(2), 441–444 (2000) https://doi.org/10.1103/PhysRevLett.85.441

[9] Scarani, V., Bechmann-Pasquinucci, H., Cerf, N.J., Dušek, M., Lütkenhaus, N., Peev, M.: The security of practical quantum key distribution. Rev. Mod. Phys. **81**(3), 1301–1350 (2009) https://doi.org/10.1103/RevModPhys.81.1301

[10] Okuła, R., Mironowicz, P.: How decoherence affects the security of BB84 quantum key distribution protocol. Quantum Inf. Process. **24**(2) (2025) https://doi.org/10.1007/s11128-025-04650-8

[11] Javadi-Abhari, A., Treinish, M., Krsulich, K., Wood, C.J., et al.: Quantum computing with Qiskit (2024)

[12] Ahmed, Z.: BB84 Quantum Key Distribution on PKTron and IBM Quantum Hardware: Eavesdropper-Detection Validation for Secure Military Communications. Preprint, Research Square (2026). https://doi.org/10.21203/rs.3.rs-10838323/v1

[13] Pathare, A., Patil, P., Pimple, A., Sawant, V.: Experimental Analysis on the BB84 Quantum Key Distribution Protocol Using Qiskit and IBM Quantum Hardware. Preprint, TechRxiv (2025). https://doi.org/10.36227/techrxiv.173933577.75659816/v1

[14] Mueller, R., Ribezzo, D., Zahidy, M., Oxenløwe, L.K., Bacco, D., Forchhammer, S.: Efficient information reconciliation for high-dimensional quantum key distribution. Quantum Inf. Process. **23**(5) (2024) https://doi.org/10.1007/s11128-024-04395-w

[15] Duan, X., Li, B., Yin, Z., Zhang, X., Luo, B.: Robust image steganography against lossy JPEG compression based on embedding domain selection and adaptive error correction. Expert Syst. Appl. **229**, 120416 (2023) https://doi.org/10.1016/j.eswa.2023.120416

[16] Zhang, J., He, X., Cao, Y.: Errorless robust JPEG steganography using steganographic polar codes. EURASIP J. Inf. Secur. **2024**(1) (2024) https://doi.org/10.1186/s13635-024-00173-4

[17] Chen, B., Wornell, G.W.: Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. IEEE Trans. Inf. Theory **47**(4), 1423–1443 (2001) https://doi.org/10.1109/18.923725

[18] Zhang, Y., Luo, X., Yang, C., Ye, D., Liu, F.: A framework of adaptive steganography resisting JPEG compression and detection. Secur. Commun. Netw. **9**(15), 2957–2971 (2016) https://doi.org/10.1002/sec.1502

[19] Jaya Lakshmi Narayana, B., Bala Sumanth Reddy, M., Sathvik, K., Balavignesh Reddy, K.: Quantum blind image watermarking with QDCT-QIM and BB84 key distribution. EPJ Web Conf. **360**, 01024 (2026) https://doi.org/10.1051/epjconf/202636001024

[20] Zhang, Y., Lu, K., Gao, Y., Wang, M.: NEQR: a novel enhanced quantum representation of digital images. Quantum Inf. Process. **12**(8), 2833–2860 (2013) https://doi.org/10.1007/s11128-013-0567-z

[21] Jiang, N., Zhao, N., Wang, L.: LSB based quantum image steganography algorithm. Int. J. Theor. Phys. **55**(1), 107–123 (2015) https://doi.org/10.1007/s10773-015-2640-0

[22] Wang, M.-X., Yang, H.-M., Jiang, D.-H., Yan, B., Pan, J.-S., Liu, T.: A novel quantum color image steganography algorithm based on turtle shell and LSB. Quantum Inf. Process. **21**(4) (2022) https://doi.org/10.1007/s11128-022-03494-w

[23] Dang, H.-M., Yang, H.-M., Jiang, D.-H., Yan, B., Huang, J.-H., Sun, X.-T., Yang, X.-H.: A LSB quantum steganography algorithm based on hash encryption.

Quantum Inf. Process. **24**(7) (2025) https://doi.org/10.1007/s11128-025-04818-2

[24] Iranmanesh, S., Atta, R., Ghanbari, M.: Implementation of a quantum image watermarking scheme using NEQR on IBM quantum experience. Quantum Inf. Process. **21**(6) (2022) https://doi.org/10.1007/s11128-022-03530-9

[25] Tudorache, A.-G., Manta, V., Caraiman, S.: Quantum steganography based on the B92 quantum protocol. Mathematics **10**(16), 2870 (2022) https://doi.org/10.3390/math10162870

[26] Clopper, C.J., Pearson, E.S.: The use of confidence or fiducial limits illustrated in the case of the binomial. Biometrika **26**(4), 404–413 (1934) https://doi.org/10.1093/biomet/26.4.404

[27] Brassard, G., Salvail, L.: Secret-key reconciliation by public discussion. In: Advances in Cryptology — EUROCRYPT '93. Lecture Notes in Computer Science, pp. 410–423. Springer, Berlin, Heidelberg (1994). https://doi.org/10.1007/3-540-48285-7_35

[28] Carter, J.L., Wegman, M.N.: Universal classes of hash functions. J. Comput. Syst. Sci. **18**(2), 143–154 (1979) https://doi.org/10.1016/0022-0000(79)90044-8

[29] Bennett, C.H., Brassard, G., Crépeau, C., Maurer, U.M.: Generalized privacy amplification. IEEE Trans. Inf. Theory **41**(6), 1915–1923 (1995) https://doi.org/10.1109/18.476316

[30] Impagliazzo, R., Levin, L.A., Luby, M.: Pseudo-random generation from one-way functions. In: Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing (STOC '89), pp. 12–24. ACM Press, ??? (1989). https://doi.org/10.1145/73007.73009

[31] Tomamichel, M., Lim, C.C.W., Gisin, N., Renner, R.: Tight finite-key analysis for quantum cryptography. Nat. Commun. **3**(1) (2012) https://doi.org/10.1038/ncomms1631

[32] Krawczyk, H., Eronen, P.: HMAC-based extract-and-expand key derivation function (HKDF). Request for Comments 5869, RFC Editor (2010). https://doi.org/10.17487/RFC5869

[33] National Institute of Standards and Technology: Advanced encryption standard (AES). Federal Information Processing Standards Publication 197 (updated), National Institute of Standards and Technology (2023). https://doi.org/10.6028/NIST.FIPS.197-upd1

[34] Reed, I.S., Solomon, G.: Polynomial codes over certain finite fields. J. Soc. Ind. Appl. Math. **8**(2), 300–304 (1960) https://doi.org/10.1137/0108018

[35] Mallat, S.G.: A theory for multiresolution signal decomposition: the wavelet representation. IEEE Trans. Pattern Anal. Mach. Intell. **11**(7), 674–693 (1989) https://doi.org/10.1109/34.192463

[36] Wallace, G.K.: The JPEG still picture compression standard. IEEE Trans. Consum. Electron. **38**(1), (1992) https://doi.org/10.1109/30.125072

[37] Bas, P., Filler, T., Pevný, T.: "Break Our Steganographic System": The ins and outs of organizing BOSS. In: Information Hiding. Lecture Notes in Computer Science, pp. 59–70. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24178-9_5

[38] Eastman Kodak Company: Kodak Lossless True Color Image Suite. https://r0k.us/graphics/kodak/. Accessed 28 October 2026

[39] Wilson, E.B.: Probable inference, the law of succession, and statistical inference. J. Am. Stat. Assoc. **22**(158), 209–212 (1927) https://doi.org/10.1080/01621459.1927.10502953

[40] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004) https://doi.org/10.1109/TIP.2003.819861

[41] Westfeld, A., Pfitzmann, A.: Attacks on steganographic systems. In: Information Hiding. Lecture Notes in Computer Science, pp. 61–76. Springer, Berlin, Heidelberg (2000). https://doi.org/10.1007/10719724_5

[42] Fridrich, J., Goljan, M., Du, R.: Reliable detection of LSB steganography in color and grayscale images. In: Proceedings of the 2001 Workshop on Multimedia and Security: New Challenges (MM&Sec '01), p. 27. ACM Press, ??? (2001). https://doi.org/10.1145/1232454.1232466

[43] Pevný, T., Bas, P., Fridrich, J.: Steganalysis by subtractive pixel adjacency matrix. IEEE Trans. Inf. Forensics Secur. **5**(2), 215–224 (2010) https://doi.org/10.1109/TIFS.2010.2045842

[44] Fridrich, J., Kodovský, J.: Rich models for steganalysis of digital images. IEEE Trans. Inf. Forensics Secur. **7**(3), 868–882 (2012) https://doi.org/10.1109/TIFS.2012.2190402