STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **16**, December 2026, pp 5831–5846.
Published online in International Academic Press (www.IAPress.org)

Self-Synchronising Timestamp-Keyed Encryption for LSB Image Steganography with In-Image Key Transport

Muhammad Rifqi [1,*], Hadhrami Ab Ghani [2], Hasyiya Karimah[2], Hadi Santoso[1]

[1] *Faculty of Computer Science, Universitas Mercu Buana, Indonesia*
[2] *Faculty of Data and Computing Sciences, Universiti Malaysia Kelantan, Malaysia*

Abstract

Background: Hybrid schemes that combine lightweight encryption with image steganography are attractive for resource-constrained environments, but time-derived keys introduce an operational problem that is rarely addressed: sender and receiver must agree on the exact key material, and clock drift of a single second causes decryption to fail.
Objective: This study proposes a timestamp-keyed substitution cipher whose key material travels inside the stego image itself, removing the need for clock synchronisation or a separate key channel while retaining session-dependent ciphertext variability.
Methods: A position-dependent shift sequence is derived from timestamp components combined with a pre-shared key. The timestamp is written into a reserved header region of the cover image together with the ciphertext length, so that the receiver recovers the shift sequence directly from the stego image. Embedding uses single-bit least-significant-bit substitution on the 24-bit RGB pixel data of a BMP cover. The two key components serve distinct roles: the timestamp provides inter-session variability and is transmitted openly, whereas the pre-shared key provides confidentiality and is never transmitted.
Results: Across five 512×512 USC-SIPI cover images, the scheme achieves a mean PSNR of 77.8575 ± 0.0699 dB and an NPCR of $0.106506 \pm 0.00172\%$ at a payload of 199 characters (0.002116 bpp), with message-level BER of zero across all tested payloads from 50 to 2000 characters, confirming lossless recovery. Ciphertext variability across different transmission times reaches 60.30%, compared with 0.00% for a fixed-shift Caesar baseline, and the byte-frequency entropy rises from 4.2417 to 5.8035. Chi-square and RS analysis do not separate the stego images from their covers, the R_M and R_{-M} curves differing by 0.071 percentage points. The complete pipeline executes in 3.606 ± 0.283 ms per image.
Originality/Value: The contribution is the in-image transport of time-derived key material, which makes a time-keyed cipher deployable without synchronised clocks. The roles of the transmitted timestamp and the withheld pre-shared key are separated explicitly, and the security claims are bounded accordingly.
Conclusion: The scheme offers a reproducible, self-contained mechanism for lightweight confidential data hiding. Its limitations, including the periodicity of the cipher and the absence of validation against learning-based steganalysis, are stated explicitly.

Keywords Image Steganography, Least Significant Bit, Timestamp-Based Key Generation, Key Transport, Image Quality Assessment, Steganalysis

AMS 2010 subject classifications 68P25, 94A60, 68U10

DOI: 10.19139/soic-2310-5070-4873

1. Introduction

Digital images are an attractive carrier for confidential communication because they are exchanged in enormous volume and because small perturbations of pixel values escape human notice. Least-significant-bit (LSB)

*Correspondence to: Muhammad Rifqi (Email: m.rifqi@mercubuana.ac.id). Mercu Buana University, Jalan Raya Meruya Selatan, Kembangan, West Jakarta 11650.

ISSN 2310-5070 (online) ISSN 2311-004X (print)
Copyright © 2026 International Academic Press

substitution is the oldest and simplest realisation of this idea: the low-order bit of each pixel byte is replaced by one bit of the payload, altering the byte by at most one unit [5]. The method is trivial to implement, imposes negligible computational cost, and at low embedding rates produces distortion far below any perceptual threshold.

Its weakness is equally well established. Steganography conceals the existence of a message but does not protect its content. An adversary who suspects embedding and recovers the bitstream reads the message directly. The standard remedy is to encrypt before embedding, so that extraction yields ciphertext rather than plaintext [16]. This composition is sound in principle and shifts the problem to a familiar one: the sender and the receiver must share a key.

A line of work has sought to avoid static keys by deriving key material from the time of transmission. The appeal is clear. A key that changes with every transmission ensures that identical plaintexts sent at different moments produce different ciphertexts, which denies an observer the inference that the same message was sent twice, and it limits the value of any single compromised key to a single session.

The difficulty is that a time-derived key is only usable if the receiver can reproduce the sender's time exactly. A discrepancy of one second changes every element of the key stream and decryption fails completely. Three responses are available, and none is satisfactory in the setting steganography is meant to serve. Synchronised clocks impose an infrastructural requirement that a covert channel is often chosen precisely to avoid. A tolerance window, the approach taken by time-based one-time password schemes [14], requires the receiver to attempt decryption across a range of candidate times and to possess a reliable criterion for recognising success. A separate key channel reintroduces exactly the key-distribution problem that time-derived keys were introduced to remove.

This paper takes a different route. The timestamp is written into a reserved header region of the stego image itself, at a fixed position and of fixed length, ahead of the encrypted payload. The receiver reads the header, obtains the timestamp, combines it with the pre-shared key to regenerate the key stream, and decrypts. No clock synchronisation is required, no candidate times are tried, and no second channel is opened. The stego image is self-contained: it carries both the message and the information needed to interpret it.

The distinction is worth stating precisely, since it is the whole of the contribution. The timestamp is not a secret and is not treated as one. It is a public nonce that guarantees session-to-session variability. Confidentiality rests entirely on the pre-shared key, which never travels with the image. What the header transports is the synchronisation information, not the secret.

Research Questions

RQ1. To what extent does a timestamp-derived, position-dependent shift sequence increase ciphertext variability and symbol-distribution entropy relative to a fixed-shift cipher?

RQ2. Can the key material required for decryption be transported inside the stego image without measurable cost in image quality, and does recovery remain exact across payload sizes?

Contributions

1. A key transport mechanism in which the timestamp governing encryption is embedded in a fixed-length header region of the stego image, eliminating the clock-synchronisation requirement that otherwise makes time-derived keys impractical.
2. A timestamp-keyed polyalphabetic substitution cipher whose shift sequence depends jointly on the transmission time, a pre-shared key, and byte position.
3. An evaluation on five standard USC-SIPI covers [19] reporting image quality, capacity–distortion behaviour, ciphertext variability, statistical preservation, chi-square [20] and RS [8] steganalysis, and stage-wise timing, with all measurements produced by a single reproducible execution.
4. An explicit security analysis that identifies the limits of the construction, including the periodicity of the cipher and the known-plaintext foothold created by the header itself.

The remainder of the paper is organised as follows. Section 2 reviews related work. Section 3 describes the proposed scheme. Section 4 reports the experimental results. Section 5 discusses their interpretation and the limitations of the work, and Section 6 concludes.

2. Related Work

2.1. Spatial-Domain Steganography

LSB substitution in the spatial domain was formalised by Chan and Cheng [5], who analysed the worst-case distortion of k-bit substitution and introduced an optimal pixel adjustment process to reduce it. For $k = 1$, the case considered in this paper, the expected squared error per modified byte is 0.5, which places the method at the low-distortion end of the spatial-domain family.

Pixel-value differencing, introduced by Wu and Tsai [21], varies the embedding capacity with local texture: pairs of adjacent pixels with a large difference are assumed to lie in a busy region and are allotted more bits. The method raises capacity substantially over uniform LSB while keeping distortion perceptually acceptable, and it has generated a large body of derivative work. Turakulova et al. [18] combine PVD with an XOR-based variable LSB stage, and Fayaz et al. [7] apply dynamic programming to the allocation of bits across pixel pairs.

These adaptive schemes pursue a different objective from the present work. They maximise capacity subject to a distortion constraint. The scheme described here operates at a deliberately low embedding rate and addresses key management rather than capacity, so adaptive allocation is orthogonal to it: the key transport mechanism of Section 3.3 could be applied on top of an adaptive embedder without modification. We adopt uniform single-bit LSB because it makes the distortion analytically predictable, which allows the measured results to be checked against theory.

Mielikainen [12] showed that embedding two bits per pixel pair while modifying at most one pixel reduces the expected number of changes relative to LSB matching, an illustration of the general principle that steganographic security improves as the number of modifications falls. That principle bears on a result reported in Section 4.4, where encryption is found to bring the number of modified bytes closer to its theoretical expectation.

Adaptive allocation need not rely on fixed thresholds. Yadav et al. [22] apply an interval type-1 fuzzy logic system to decide how many bits each pixel may carry, letting the embedding capacity vary continuously with local image characteristics rather than switching abruptly between discrete modes. Their approach shares with the adaptive methods above an objective the present work does not pursue, namely maximising capacity, but it illustrates how far the embedding decision itself can be refined once that is the goal.

2.2. Encryption Combined with Embedding

Encrypting before embedding is standard practice, and the literature differs mainly in the choice of cipher. Chaos-based constructions are common: Al-Rekaby et al. [1] use chaotic maps to scramble video frames prior to embedding, exploiting sensitivity to initial conditions to produce key-dependent diffusion. Mansouri et al. [11] combine DNA computing with chaotic sequences for image encryption, reporting large key spaces and strong statistical properties.

Such schemes are effective at destroying the statistical structure of the payload, but they inherit the key-distribution assumption without examining it. The key, or the parameters from which it is derived, must reach the receiver by some means the paper does not specify. Where the key is derived from transmission time, this assumption becomes an active problem rather than a background one, because the receiver must reproduce a value that changes continuously.

Amine et al. [2] combine a two-dimensional Hénon chaotic map with a Josephus-ZigZag permutation to encrypt colour images directly, reporting both strong statistical properties and low computational cost. Their emphasis on execution speed parallels the positioning of the present work for resource-constrained deployment, though they encrypt the image itself rather than a payload embedded within it.

2.3. Time-Derived Keys and the Synchronisation Problem

The problem of deriving a shared secret from time is treated rigorously outside the steganography literature, in the design of one-time password systems. HOTP [13] derives a one-time value from a shared secret and a monotonic counter. TOTP [14] replaces the counter with a time step, typically thirty seconds, so that both parties derive the same value from a shared secret and a common clock.

RFC 6238 addresses clock drift explicitly, and its treatment is instructive. The specification requires that the validator accept values from a configurable number of time steps either side of its own, and it discusses the resulting trade-off: a wider window tolerates more drift but enlarges the interval during which a captured value remains valid. The tolerance window is a managed compromise, not a solution, and it is workable in the OTP setting only because the space of candidate values is small and success is immediately recognisable.

Neither condition holds for a timestamp-keyed stream cipher applied to an embedded payload. A one-second resolution over even a modest window yields many candidate key streams, and the receiver recovering arbitrary binary data has no reliable criterion for deciding which trial decryption is correct. Transporting the timestamp alongside the ciphertext removes the need for any such criterion.

An earlier formulation by the present authors [17] derived a substitution key from transmission-time components but required both parties to agree on that timestamp in advance, and therefore inherited the synchronisation requirement described above. The present work removes that requirement.

The construction of keystreams from structured mathematical objects offers a further route to session-dependent key material. Dafik et al. [6] derive a keystream from hyperladder graph labellings for text encryption and document watermarking, obtaining a key sequence whose period is governed by the graph rather than by a fixed secret. Such constructions share the present concern with generating variable key material, but they leave open the same question addressed here: how the receiver comes to possess the parameters that generated it.

The idea of carrying synchronisation information with the data it governs is not itself novel; initialisation vectors are transmitted in clear alongside ciphertext in standard block-cipher modes for exactly this reason. The contribution here is the observation that in steganography the cover provides a natural location for such a field, so the synchronisation information can be carried without any externally visible change to the communication, and the demonstration that doing so costs nothing measurable in image quality.

2.4. Steganalysis

Any claim of undetectability is meaningful only against named detectors. Two first-generation attacks target sequential LSB embedding specifically.

The chi-square attack of Westfeld and Pfitzmann [20] exploits the fact that LSB substitution equalises the frequencies of the two members of each pair of values $\{2i, 2i+1\}$. A goodness-of-fit test against the hypothesis of equal frequencies detects embedding when the rate is high, and scanning the test across increasing portions of the image localises the embedded region.

RS analysis, due to Fridrich, Goljan and Du [8], classifies groups of pixels as regular or singular under a flipping mask and exploits the asymmetry that LSB embedding introduces between the responses to the mask and its negation. The method estimates the embedding rate rather than merely signalling its presence, and remains sensitive at rates well below those the chi-square attack detects.

Both are superseded by feature-based detectors. The subtractive pixel adjacency matrix of Pevný et al. [15] models dependencies between neighbouring pixels as a Markov chain and classifies the resulting features, achieving reliable detection of LSB matching at low rates. Ker's analysis [10] established earlier that LSB matching is detectable in grayscale images by examining the histogram characteristic function. More recently, learning-based detectors such as SRNet [4] take the raw image as input and learn features directly.

We evaluate against chi-square and RS analysis, and state plainly in Sections 4.6 and 5.3 that feature-based and learning-based detection were not tested. A scheme evaluated only against first-generation detectors cannot claim resistance to current ones, and we make no such claim.

2.5. Positioning of the Present Work

Table 1 situates this work against the schemes discussed above. The relevant axis is not capacity, on which the present scheme is deliberately modest, but whether the receiver can decrypt from the stego image alone.

The final column is the point. Every time-derived scheme in the table requires the receiver to possess, in addition to the secret, some means of establishing the time the sender used. In the scheme proposed here that requirement is discharged by the stego image itself.

Table 1. Positioning relative to representative prior work.

Work	Embedding	Key source	Receiver requires
Chan & Cheng [5]	LSB with adjustment	—	—
Wu & Tsai [21]	PVD	—	—
Turakulova et al. [18]	PVD + XOR-VLSB	Static key	Pre-shared key
Al-Rekaby et al. [1]	Video LSB	Chaotic parameters	Pre-shared parameters
Mansouri et al. [11]	— (encryption only)	DNA + chaotic	Pre-shared parameters
Rifqi et al. [17]	— (encryption only)	Time + secret	Agreed timestamp
TOTP [14]	— (not steganographic)	Time + secret	Synchronised clock
This work	LSB (single bit)	Time + pre-shared key	**Pre-shared key only**

2.6. Hypotheses

The preceding review identifies a gap: schemes that derive encryption keys from time leave the synchronisation of that time between the communicating parties unspecified. The following hypotheses test whether transporting the key material inside the stego image closes that gap while preserving the properties expected of a steganographic channel.

- **H1 (Visual fidelity.)** Single-bit LSB substitution of the encrypted payload preserves the visual fidelity of the cover image, such that the induced distortion remains below the threshold of perceptual visibility, as indicated by PSNR, MSE, RMSE, MAE, RSE and NCC.
- **H2 (Lossless recovery.)** The in-image key transport mechanism permits exact recovery of the embedded message from the stego image alone, without clock synchronisation, as indicated by a message-level bit error rate of zero across all tested payload sizes.
- **H3 (Statistical undetectability.)** Embedding preserves the first- and second-order statistics of the cover image, as indicated by negligible deviation in entropy and in horizontal, vertical and diagonal pixel correlation, and does not render the stego image separable from its cover under chi-square [20] and RS [8] analysis.
- **H4 (Dynamic key effectiveness.)** Deriving the shift sequence from transmission-time components increases ciphertext variability and raises the entropy of the symbol-frequency distribution relative to a fixed-shift cipher, such that identical plaintexts encrypted at different times yield measurably different ciphertexts.

Each hypothesis is evaluated in Section 4 and the outcomes are collected in Table 2.

3. Methodology

This study proposes a self-synchronising steganographic scheme in which the key material required for decryption is transported inside the stego image. The methodology comprises four stages: derivation of a position-dependent shift sequence from transmission-time components and a pre-shared key; encryption of the secret message under that sequence; construction of a payload that carries the timestamp alongside the ciphertext; and embedding of that payload into the cover image by single-bit least-significant-bit substitution.

3.1. Research Framework

The framework consists of two sequential layers. The first transforms the plaintext into ciphertext under a shift sequence that varies with symbol position. The second embeds the resulting ciphertext, preceded by a header carrying the timestamp and the payload length, into the pixel data of a lossless cover image.

The design addresses an operational gap in prior time-keyed schemes. When the key is derived from transmission time and that time is not transmitted, the receiver must reproduce it independently, which requires synchronised clocks and a tolerance window; a discrepancy of one second causes decryption to fail entirely. By writing the timestamp into the stego image, the proposed scheme removes this requirement: the receiver reads the timestamp from the image it has already received.

This choice has a direct security consequence, which is stated here rather than deferred. A timestamp transmitted in the open provides no confidentiality by itself. Its function is to ensure that identical plaintexts sent at different times produce different ciphertexts and different stego images. Confidentiality rests on the pre-shared key, which is never transmitted. Section 3.5 quantifies the resulting key space.

3.2. Timestamp-Keyed Cipher

Let the transmission time be represented as an ordered sequence of integer components

$$T = [t_1, t_2, \ldots, t_m], \tag{1}$$

where t_j denotes the j-th component of the selected timestamp representation and m the number of components used. Let $K = [k_1, k_2, \ldots, k_n]$ denote a pre-shared key of n integer components.

The shift applied to the symbol at position i is

$$s_i = \left(t_{(i \bmod m)+1} + k_{(i \bmod n)+1}\right) \bmod 256. \tag{2}$$

Encryption and decryption operate on the byte representation of the message,

$$c_i = (p_i + s_i) \bmod 256, \qquad p_i = (c_i - s_i) \bmod 256, \tag{3}$$

where p_i and c_i denote the i-th plaintext and ciphertext byte.

Operating on bytes rather than on alphabetic characters alone is deliberate. A cipher that leaves punctuation, digits and whitespace unchanged preserves the structural outline of the message, which is exploitable independently of the substitution applied to the letters. Equation (3) transforms every byte of the payload, so no such outline survives.

The period of the shift sequence is $\operatorname{lcm}(m, n)$. Choosing m and n coprime maximises this period for given component counts; with $m = 5$ and $n = 3$, the configuration used throughout this study, the sequence repeats every 15 symbols.

3.3. In-Image Key Transport

The payload written into the cover image is

$$M = H_T \,\|\, H_L \,\|\, C, \tag{4}$$

where C is the ciphertext, H_T is a fixed-length field of ν bytes carrying the timestamp components, and H_L is a four-byte big-endian field carrying $|C|$. With $\nu = 5$ the header occupies $8(\nu + 4) = 72$ bits.

The receiver reads H_T and H_L first, from the leading $8(\nu + 4)$ pixel bytes in raster order. The recovered timestamp, combined with the pre-shared key, regenerates the shift sequence of Equation (2); the recovered length delimits the ciphertext. No sentinel value is required, and no assumption is made about the receiver's clock.

Two properties follow. First, the scheme is self-synchronising: decryption succeeds regardless of when the stego image is received or in which time zone. Second, the header is fixed in position and length, so its recovery does not depend on any quantity that the header itself encodes.

3.4. Embedding and Extraction

The cover image is a 24-bit BMP file. Let I denote its pixel byte array, obtained by skipping the file header at the offset recorded in the BMP structure. Embedding replaces the least significant bit of consecutive pixel bytes with the bits of M in raster order,

$$I'_i = (I_i \wedge \texttt{0xFE}) \vee b_i, \qquad 0 \le i < 8|M|, \tag{5}$$

where b_i denotes the i-th bit of M and $\wedge, \vee$ denote bitwise conjunction and disjunction. Bytes beyond $8|M|$ are copied unchanged.

Because Equation (5) modifies only bit 0, the per-byte error is constrained to $\{0, 1\}$, and under the assumption of uniformly distributed payload bits the expected squared error per modified byte is $1/2$. This accounts for the numerical coincidence of MAE and MSE reported in Section 4.2: the two differ in unit but not in value when the error is binary.

Algorithm 1: Embedding with In-Image Key Transport

Input: Plaintext P; timestamp T; pre-shared key K; cover BMP I
Output: Stego BMP I_s
$S \leftarrow \text{ShiftSequence}(|P|, T, K)$; // Equation (2)
$C \leftarrow \text{Encrypt}(P, S)$; // Equation (3)
$H_T \leftarrow \text{EncodeTimestamp}(T)$;
$H_L \leftarrow \text{EncodeLength}(|C|)$;
$M \leftarrow H_T \,\|\, H_L \,\|\, C$; // Equation (4)
$(\text{header}, I) \leftarrow \text{SplitBMP}(I)$;
if $8|M| > |I|$ **then return** "Insufficient capacity";
for $i \leftarrow 0$ **to** $8|M| - 1$ **do**
  $I_i \leftarrow (I_i \wedge \texttt{0xFE}) \vee M[i]$;
return header $\|\, I$;

Algorithm 2: Extraction and Decryption

Input: Stego BMP I_s; pre-shared key K
Output: Plaintext P
$(\cdot, I) \leftarrow \text{SplitBMP}(I_s)$;
$H \leftarrow \text{ReadBits}(I, 0, 8(\nu + 4))$;
$T \leftarrow \text{DecodeTimestamp}(H)$; $L \leftarrow \text{DecodeLength}(H)$;
$C \leftarrow \text{ReadBits}(I, 8(\nu + 4), 8L)$;
$S \leftarrow \text{ShiftSequence}(L, T, K)$;
return $\text{Decrypt}(C, S)$;

The two procedures are exact inverses: the header is read from a fixed position, and the shift sequence is a deterministic function of T and K alone. Section 4.1 reports a message-level bit error rate of zero across all tested payloads, which verifies this.

3.5. Key Space Analysis

The timestamp is transmitted in the open and therefore contributes no confidentiality. Its entropy is in any case modest: a representation using hour, minute and second admits $24 \times 60 \times 60 = 86{,}400$ distinct values within a day, or approximately $2^{16.4}$. An adversary who knows the approximate transmission time, which network metadata often reveals, narrows this further. The scheme therefore does not rely on the secrecy of T.

Confidentiality rests on the pre-shared key K. For n components drawn from $\{0, \ldots, 255\}$, the key space is $256^n = 2^{8n}$; with $n = 16$ this gives 2^{128}, comparable to a symmetric block cipher of the same key length. The security of the construction is nonetheless weaker than that figure suggests, because Equation (2) is a periodic polyalphabetic substitution: the shift sequence repeats with period $\text{lcm}(m, n)$, and an adversary holding sufficient known plaintext can recover the sequence by standard period-detection techniques [9].

The scheme is accordingly positioned as a lightweight obfuscation layer for resource-constrained deployment, not as a replacement for a modern block cipher. Where the threat model includes an adversary with known-plaintext access, a standard lightweight cipher [3] should be substituted for Equation (3); the in-image key transport mechanism of Section 3.3 is independent of the cipher used and applies unchanged.

3.6. Experimental Setup

Experiments use five colour images from the USC-SIPI Miscellaneous volume [19] at 512×512 pixels — Baboon (4.2.03), Lena (4.2.04), Airplane (4.2.05), Sailboat (4.2.06) and Peppers (4.2.07) — obtained as TIFF from the source archive and converted to 24-bit BMP without resampling or recompression. Payloads of 50, 100, 250, 500, 1000 and 2000 characters are tested on each image, in addition to a 199-character reference payload used throughout Section 4.

For reproducibility, all reported results use the fixed reference timestamp $T = [14, 39, 41, 48, 37]$ and pre-shared key $K = [7, 13, 3]$, giving $m = 5$, $n = 3$ and a shift period of 15 bytes. The scheme is deterministic given these values. Results are reported as the mean and standard deviation across the five cover images.

All metrics are computed on the full 24-bit RGB pixel data rather than on a grayscale projection. This distinction matters: embedding modifies the red, green and blue bytes independently, and a luminance projection mixes them, so metrics computed after conversion do not describe the distortion actually introduced.

Measurements were performed on a Google Colab instance with two virtual CPU cores of an Intel Xeon processor at 2.20 GHz and 12 GiB of RAM, running Ubuntu 24.04.5 LTS with Python 3.13.15, NumPy 2.1.3 and Pillow 11.3.0. Timing figures are the mean over 30 runs, preceded by three warm-up iterations that were discarded, using `time.perf_counter()`.

4. Results

This section reports the experimental results. All figures derive from a single execution of the implementation on the five cover images described in Section 3.6, with the fixed timestamp and pre-shared key stated there. Unless noted otherwise, values are means over the five images with standard deviations, computed on the full 24-bit RGB pixel data rather than on a luminance projection.

4.1. Message Recovery and Key Transport

The scheme recovered the embedded message exactly from all five stego images. In each case the receiver read the timestamp from the reserved header region, regenerated the shift sequence from that timestamp and the pre-shared key, and decrypted the payload without any knowledge of when the image had been produced. The message-level bit error rate is zero (Table 3), and remains zero at every payload size tested (Table 4).

This result establishes the central property of the scheme. A time-derived key is normally unusable without synchronised clocks, since a discrepancy of one second changes every shift in the sequence. Transporting the timestamp inside the stego image removes that requirement entirely: no tolerance window, no trial decryption over candidate times, and no separate key channel. Hypothesis H2 is therefore supported.

4.2. Visual Fidelity

Table 3 reports the image-quality metrics at the reference payload of 199 characters (0.002116 bpp). The mean PSNR is 77.8575 ± 0.0699 dB, nearly twice the 40 dB threshold conventionally associated with imperceptible embedding, and the standard deviation across five structurally very different covers is below 0.07 dB, indicating that the distortion depends on the payload rather than on image content.

MSE and MAE coincide numerically at 1.0651×10^{-3}. This is not an error but a consequence of single-bit substitution: the per-byte error is confined to $\{0, 1\}$, where $x^2 = x$. The two quantities differ in unit but not in value under this constraint. The verification that exactly one bit changes per modified byte, reported by the implementation, confirms that no higher-order bit is disturbed.

NCC reaches 0.999999975, and the relative squared error is of order 10^{-8}. Hypothesis H1 is supported.

4.3. Capacity–Distortion Trade-off

Table 4 reports quality as a function of payload across six sizes from 50 to 2000 characters, corresponding to 0.000600–0.020437 bpp. PSNR declines monotonically from 83.3625 dB to 68.0359 dB, and NPCR rises from

0.030009% to 1.022110%, both approximately linearly in the number of embedded bits, as single-bit substitution predicts. Recovery is exact at every size.

The capacity is deliberately low. At 0.002116 bpp the reference payload occupies roughly one fifth of one percent of the single-bit LSB capacity of the cover. The scheme is therefore not competitive with adaptive methods that reach 1–3 bpp, and the present work makes no claim in that direction. What the curve establishes is that the scheme degrades predictably and recovers reliably across a range spanning a factor of forty in payload, which answers RQ2.

4.4. Comparison with Baselines

Table 5 compares three variants at an identical payload: plain LSB without encryption, a fixed-shift Caesar cipher followed by LSB, and the proposed scheme.

The three are statistically indistinguishable in image quality. The difference between plain LSB (78.0273 ± 0.24 dB) and the proposed scheme (77.8575 ± 0.07 dB) is 0.17 dB, well within sampling variation ($t = 1.52$, $df = 4$, not significant at $\alpha = 0.05$). This is the expected outcome: all three embed the same number of bits, and encryption operates before embedding rather than during it. The encryption layer is therefore free in terms of fidelity.

The count of modified bytes is more informative. For payload bits drawn uniformly at random, half the touched bytes should change, giving an expected 832.0 modified bytes at this payload. The proposed scheme modifies 837.6, plain LSB 806.4, and the static cipher 823.6. The proposed scheme is closest to the theoretical value because its ciphertext approaches a uniform byte distribution, whereas unencrypted ASCII text carries a non-uniform bit pattern that systematically biases the number of flips.

This observation matters beyond bookkeeping. LSB substitution is hardest to detect when the embedded bitstream is indistinguishable from random, since statistical detectors exploit departures from that condition. Encryption in this scheme therefore contributes to steganographic security as well as to confidentiality.

4.5. Ciphertext Variability

Table 5 also reports ciphertext variability, measured as the mean pairwise difference between ciphertexts of the same plaintext encrypted at five different transmission times ($n = 10$ pairs).

The fixed-shift cipher produces a variability of 0.00%: identical plaintexts yield byte-identical ciphertexts regardless of when they are sent, and therefore byte-identical stego images. An observer comparing two transmissions learns immediately that the same message was sent twice. The proposed scheme reaches 60.30%.

The effect is also visible in the symbol distribution. The byte-frequency entropy of the plaintext is 4.2417 bits. A fixed shift leaves this unchanged at 4.2417 bits, since translating a histogram does not alter its shape, and this is precisely why classical frequency analysis defeats monoalphabetic substitution. The position-dependent shift raises the entropy to 5.8035 bits by distributing each plaintext symbol across multiple ciphertext values.

These two measurements answer RQ1 and support H4. We note their limits: higher entropy and inter-session variability raise the cost of frequency analysis but do not make the construction resistant to an adversary with known plaintext, as Section 3.5 discusses.

4.6. Statistical Preservation and Steganalysis

The statistical block of Table 3 compares cover and stego directly. Entropy rises from 7.364957 to 7.365514 bits, a deviation of 5.6×10^{-4}, consistent with the injection of a small quantity of near-uniform bits. Horizontal, vertical and diagonal correlations are unchanged to six decimal places (0.952945, 0.937259 against 0.937254, and 0.911972 against 0.911967), indicating that the spatial dependency structure of the cover survives embedding.

Figure 1 presents the imperceptibility analysis on the Baboon image. The byte-value histograms of cover and stego overlap throughout the range (panel a), and the signed difference (panel b) confirms that embedding only redistributes counts between adjacent values, with no displacement across the histogram. Panel (d) reports the pair-of-values imbalance that the chi-square attack exploits: the mean falls from 0.03676 to 0.03437, a change too small to separate the two distributions.

Figure 2 reports the steganalysis. In panel (a) the chi-square p-value is high for small analysed portions, where too few pair-of-values bins carry sufficient counts for the statistic to be meaningful, then falls below the detection threshold at approximately 25% of the image and reaches zero thereafter. The decisive observation is that the cover and stego curves are superimposed over the entire range: the attack responds identically to an image that carries a payload and one that does not.

Panel (b) shows the RS diagram. At zero additional flipping the R_M and R_{-M} curves differ by 0.071 percentage points, and the four curves follow the pattern expected of a cover with a very low embedding rate. Hypothesis H3 is supported for these two detectors.

The behaviour of the chi-square curve on the cover image itself warrants comment. The p-value of the cover also falls below the detection threshold beyond approximately 25% of the image, which would nominally indicate embedding in an image that carries none. This reflects a known limitation of the attack rather than a property of the scheme: the test assumes that the two members of each pair of values are unequally populated in a clean cover, an assumption that fails for highly textured images whose byte-value histogram is locally flat. Baboon is such an image. The attack is therefore uninformative on this cover in either direction, and the relevant observation is the one stated above: the cover and stego curves coincide, so the embedding introduces no pair-of-values signature that the test can isolate. A cover with stronger histogram structure would give the attack more purchase, and evaluating the scheme on such covers is left to future work.

We state the boundary of this result explicitly. Chi-square and RS analysis are first-generation detectors. The scheme has not been evaluated against feature-based detectors such as SPAM [15] or against learning-based steganalysis [4], and no claim of resistance to those is made here.

4.7. Computational Performance

Table 6 reports execution time over 30 runs, preceded by three discarded warm-up iterations. The complete pipeline, comprising encryption, embedding, extraction and decryption, executes in 3.606 ± 0.283 ms on a 512×512 cover.

The distribution across stages is informative. Encryption accounts for 0.102 ms, or 2.8% of the total, while embedding accounts for 2.524 ms and extraction with decryption for 0.980 ms. The cryptographic layer is thus a negligible fraction of the cost; the expense lies in traversing the pixel array. This supports the positioning of the scheme for resource-constrained deployment, though we note that the measurement is of a Python implementation and that a compiled implementation would be substantially faster.

4.8. Hypothesis Outcomes

Table 2. Mapping of hypotheses to supporting evidence.

Hypothesis	**Metrics**	**Evidence**	**Outcome**
H1 (visual fidelity)	PSNR, MSE, RMSE, MAE, RSE, NCC	Table 3	Supported
H2 (lossless recovery)	Message-level BER	Tables 3, 4	Supported
H3 (statistical undetectability)	Entropy, correlation, χ^2, RS	Table 3, Figs. 1, 2	Supported*
H4 (dynamic key effectiveness)	Ciphertext variability, entropy	Table 5	Supported

* Supported with respect to chi-square and RS analysis. Feature-based and learning-based steganalysis were not evaluated; see Section 5.3.

5. Discussion

5.1. Interpretation of the Results

The contribution of this work is narrow and specific: the key material required to decrypt a message travels inside the stego image that carries the message. The experimental results are best read against that claim rather than against the broader question of whether LSB substitution is a good embedding strategy.

Table 3. Image-quality evaluation over five 512×512 USC-SIPI cover images at a 199-character payload (0.002116 bpp). Values are mean $\pm$ standard deviation, computed on the full 24-bit RGB pixel data.

Category	Metric	Value	Ideal / Reference
Visual fidelity	PSNR (dB)	77.8575 ± 0.0699	> 40
	MSE	$(1.0651 \pm 0.0172) \times 10^{-3}$	$\to 0$
	RMSE	$(3.2634 \pm 0.0263) \times 10^{-2}$	$\to 0$
	MAE	$(1.0651 \pm 0.0172) \times 10^{-3}$	$\to 0$
	RSE	$(5.0346 \pm 1.5184) \times 10^{-8}$	$\to 0$
Similarity	NCC	$0.999999975 \pm 7.6 \times 10^{-9}$	$\to 1$
Imperceptibility	NPCR (%)	0.106506 ± 0.00172	$\to 0$
	UACI (%)	$(4.1767 \pm 0.0674) \times 10^{-4}$	$\to 0$
Statistical	Entropy (bits)	7.365514	7.364957†
	Correlation–H	0.952945	0.952945†
	Correlation–V	0.937259	0.937254†
	Correlation–D	0.911972	0.911967†
Recovery	BER (image-level)	$(1.3313 \pm 0.0215) \times 10^{-4}$	< 0.01
	BER (message-level)	0.0	0

† For entropy and correlation the reference is the cover image: the objective is preservation, not maximisation. These rows report group means without dispersion, as the cover–stego difference is the quantity of interest.

Table 4. Effect of payload size, averaged over five cover images.

Chars	Bits	bpp	PSNR (dB)	NPCR (%)	Recovered
50	472	0.000600	83.3625 ± 0.22	0.030009	✓
100	872	0.001109	80.6400 ± 0.10	0.056127	✓
250	2072	0.002635	76.9406 ± 0.10	0.131556	✓
500	4072	0.005178	73.9653 ± 0.03	0.260951	✓
1000	8072	0.010264	71.0040 ± 0.03	0.516052	✓
2000	16072	0.020437	68.0359 ± 0.03	1.022110	✓

Table 5. Comparison against baselines at an identical payload. Ciphertext variability is the mean pairwise difference between ciphertexts of the same plaintext at five transmission times ($n = 10$ pairs).

Method	PSNR (dB)	NPCR (%)	Bytes modified	Ciphertext var. (%)
Plain LSB (no encryption)	78.0273 ± 0.24	0.102539	806.4	—
Static Caesar + LSB	77.9323 ± 0.15	0.104726	823.6	0.00
Proposed (timestamp-keyed)	77.8575 ± 0.07	0.106506	837.6	60.30
Theoretical (uniform bits)	—	—	832.0	—

Table 6. Computational performance, mean over 30 runs after three discarded warm-up iterations.

Stage	Mean (ms)	Std. dev. (ms)
Encryption	0.102	0.071
Embedding	2.524	0.265
Extraction + decryption	0.980	0.147
Total	**3.606**	**0.283**

Prior schemes that derive keys from transmission time leave the receiver with a problem the literature seldom addresses. The receiver must reproduce the sender's timestamp exactly, since a discrepancy of one second changes every element of the shift sequence. In practice this forces one of three unattractive options: synchronised clocks, a tolerance window with trial decryption over candidate times, or a separate channel for the key, which reintroduces the key-distribution problem that time-derived keys were meant to avoid.

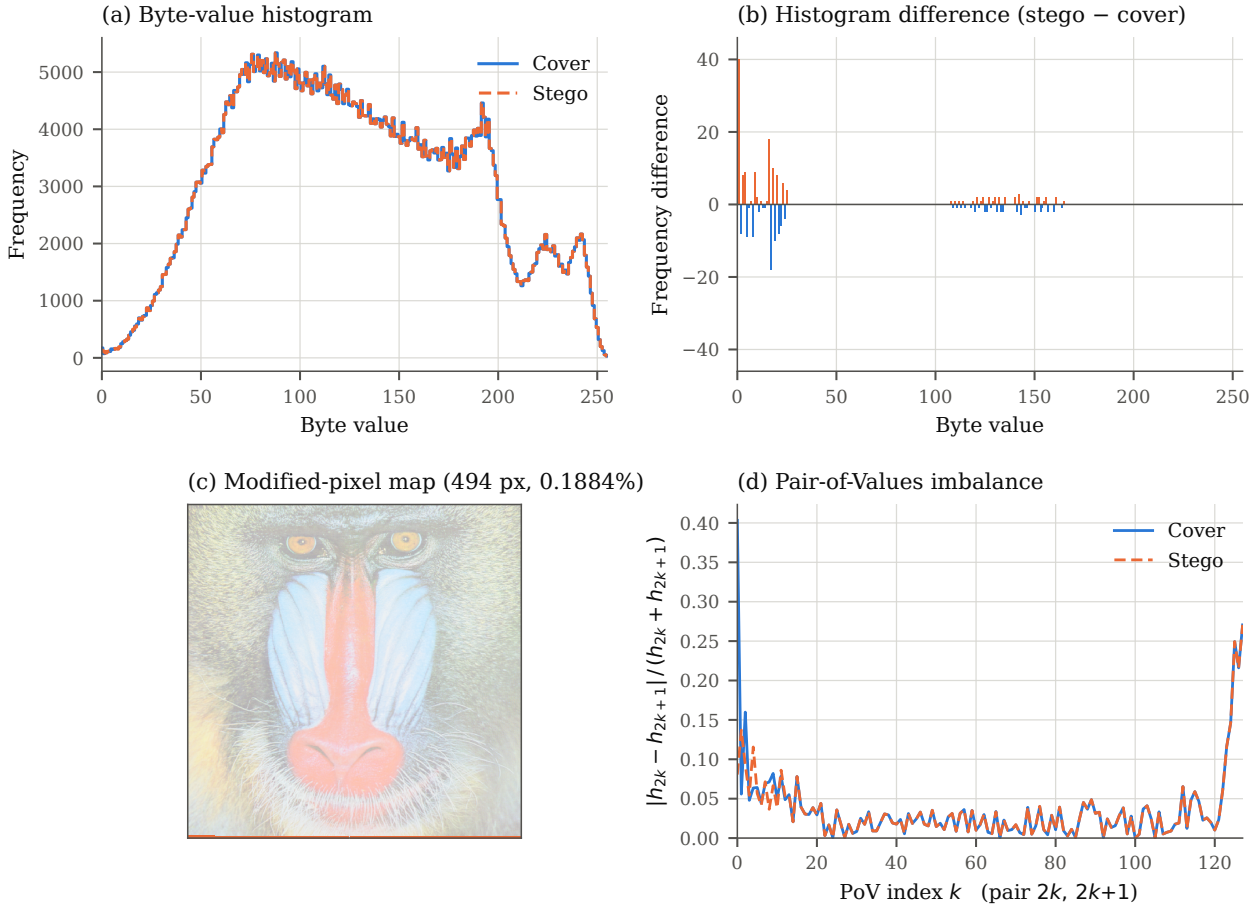


Figure 1. Imperceptibility analysis on the Baboon image (USC-SIPI 4.2.03). (a) Byte-value histograms of cover and stego overlap throughout the range. (b) The signed histogram difference shows that embedding redistributes counts only between adjacent byte values. (c) Spatial distribution of the 494 modified pixels (0.1884% of the image). (d) Pair-of-values imbalance, which the chi-square attack exploits, is essentially unchanged (0.03676 to 0.03437).

Writing the timestamp into a reserved header region of the cover removes the problem rather than managing it. The header occupies 72 bits, fixed in position and length, so its recovery depends on nothing that the header itself encodes. Message-level bit error rate is zero across five cover images and six payload sizes spanning a factor of forty, which establishes that the mechanism works rather than merely that it is plausible.

The image-quality results are unremarkable, and deliberately so. PSNR of 77.8575 ± 0.0699 dB and NPCR of 0.106506% are what single-bit LSB substitution at 0.002116 bpp should produce. The comparison in Table 5 confirms that the encryption layer costs nothing in fidelity: the three variants differ by less than 0.2 dB, a difference that is not statistically significant. The scheme neither gains nor loses image quality relative to plain LSB; what it adds is confidentiality and self-contained key transport.

One result was not anticipated. The proposed scheme modifies 837.6 bytes against a theoretical expectation of 832.0 for uniformly distributed payload bits, while plain LSB modifies 806.4. Unencrypted ASCII text carries a non-uniform bit distribution, and embedding it therefore departs from the random-payload assumption under which LSB substitution is least detectable. Encryption corrects this departure as a side effect. The security benefit of the cipher is thus not confined to the cryptographic layer; it also improves the statistical profile of the embedded bitstream.

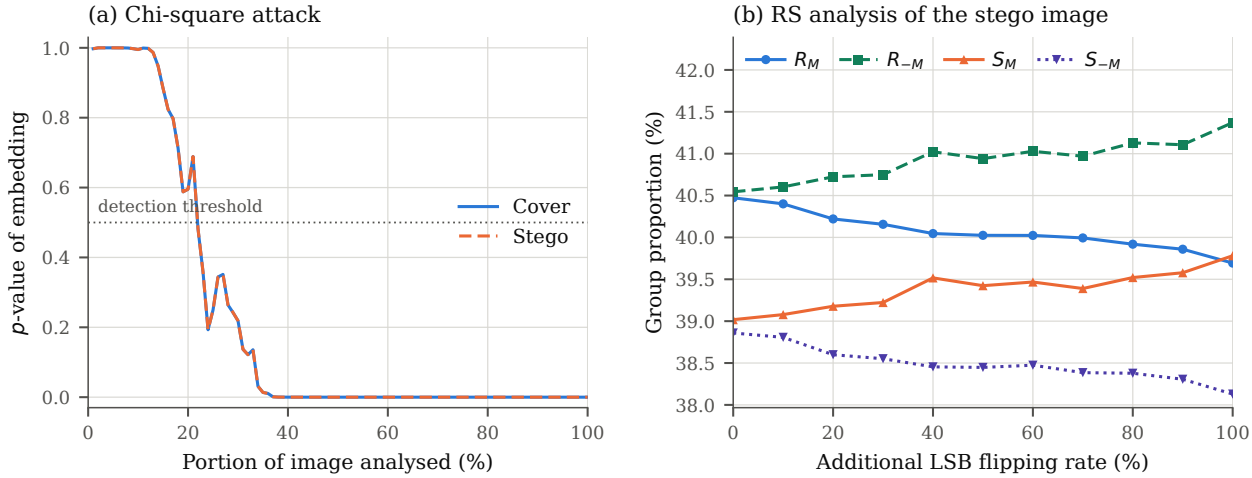


Figure 2. Steganalysis of the proposed scheme on the Baboon image. (a) Chi-square attack: the p-value of the stego image remains indistinguishable from that of the cover across the entire image, indicating that the embedding does not produce the pair-of-values equalisation the attack detects. (b) RS analysis: the R_M and R_{-M} curves differ by 0.071 percentage points at zero additional flipping, consistent with a low embedding rate.

5.2. Security Positioning

The scheme should be understood as a lightweight obfuscation layer with self-contained key transport, not as a cryptographic primitive. Three considerations bound what it provides.

First, the timestamp is transmitted openly by construction. Any party who can read the stego image can read the timestamp. Its function is to ensure that identical plaintexts sent at different times produce different ciphertexts and different stego images, which the measured variability of 60.30% against 0.00% for a fixed shift demonstrates. It is not a secret and is not treated as one.

Second, confidentiality rests entirely on the pre-shared key. The key space is 256^n for a key of n components, which is adequate for modest n, but the construction remains a periodic polyalphabetic substitution with period $\mathrm{lcm}(m, n)$. An adversary holding sufficient known plaintext can recover the shift sequence by standard period-detection techniques [9]. The rise in byte-frequency entropy from 4.2417 to 5.8035 bits raises the cost of frequency analysis but does not eliminate this vulnerability.

Third, the header is a known-plaintext foothold. The timestamp field occupies a fixed position and its components fall in predictable ranges, and the length field has a predictable structure for typical payloads. An adversary who knows the header format therefore knows a portion of the embedded bitstream exactly. The header is not encrypted, since encrypting it would require the key it exists to convey.

Where the threat model includes an adversary with known-plaintext access, the substitution cipher should be replaced by a standard lightweight block cipher [3]. The key transport mechanism described in Section 3.3 is independent of the cipher and applies unchanged; this is, in our view, the more durable part of the contribution.

5.3. Limitations

We state the limitations of this work explicitly.

Capacity. At 0.002116 bpp for the reference payload, and 0.020437 bpp at the largest tested payload, the scheme occupies a small fraction of the single-bit LSB capacity of the cover. It is not competitive with adaptive schemes that achieve 1–3 bpp, and is appropriate for short messages rather than for bulk data.

Spatial clustering of modifications. Embedding proceeds sequentially from the first pixel byte, so the modified pixels occupy a contiguous prefix of the raster order rather than a dispersed set. At the reference payload this

amounts to 494 pixels, fewer than one full row of a 512×512 cover, and the band is therefore not visually apparent at the scale of Figure 1(c). The localisation is nonetheless a property of the embedding order rather than of the payload, and a detector operating block-wise could exploit it. Distributing the payload across the cover under a key-derived permutation would address this at no cost in capacity, and is the most immediate extension of the present work.

Cipher strength. As Section 5.2 sets out, the substitution cipher is periodic and vulnerable to known-plaintext analysis, and the header provides partial known plaintext by construction.

Steganalysis coverage. Evaluation is limited to chi-square and RS analysis, both first-generation detectors. The scheme has not been evaluated against feature-based detectors such as SPAM [15], nor against learning-based steganalysis [4]. No claim of resistance to those is made.

Format dependence. The scheme requires a lossless carrier. Saving a stego image as JPEG destroys the payload entirely, since lossy quantisation does not preserve least significant bits. Robustness against compression, cropping, scaling and filtering was not evaluated.

Dataset scope. Five standard natural images at a single resolution were tested. Performance on synthetic images, medical images, or images with large uniform regions may differ; our own observation during figure preparation was that a cover with extensive flat areas concentrates the histogram perturbation into a narrow range of byte values, which a detector could plausibly exploit.

Implementation. Timing figures describe a Python implementation and should be read as an upper bound. A compiled implementation would be substantially faster, and the relative cost of the stages would likely shift.

6. Conclusion and Future Work

6.1. Conclusion

This study developed and evaluated a timestamp-keyed substitution cipher combined with least-significant-bit steganography, in which the key material is transported inside the stego image. The following conclusions follow from the experimental results.

1. **Self-contained key transport.** Embedding the timestamp in a reserved header region allows the receiver to reconstruct the shift sequence from the stego image alone. Message-level bit error rate is zero across five cover images and six payload sizes from 50 to 2000 characters, confirming that decryption succeeds without clock synchronisation, a tolerance window, or a separate key channel.
2. **Visual fidelity at no additional cost.** The scheme achieves a mean PSNR of 77.8575 ± 0.0699 dB and an NPCR of $0.106506 \pm 0.00172\%$ at 0.002116 bpp. The comparison against plain LSB and a fixed-shift baseline shows differences below 0.2 dB, not statistically significant, indicating that the encryption layer imposes no measurable penalty on image quality.
3. **Effectiveness of the dynamic key.** Identical plaintexts encrypted at different transmission times yield ciphertexts differing by 60.30% on average, against 0.00% for a fixed-shift cipher, and the byte-frequency entropy rises from 4.2417 to 5.8035 bits. A fixed shift leaves the entropy unchanged because translating a histogram does not alter its shape; the position-dependent shift distributes each plaintext symbol across multiple ciphertext values.
4. **Statistical preservation.** Entropy deviates by 5.6×10^{-4} bits and directional correlations are unchanged to six decimal places. Chi-square and RS analysis do not separate the stego images from their covers, the R_M and R_{-M} curves differing by 0.071 percentage points at zero additional flipping.

The complete pipeline executes in 3.606 ± 0.283 ms on a 512×512 cover, with the cryptographic stage accounting for 2.8% of that total.

The scheme is a lightweight mechanism for short confidential messages in settings where key distribution is the binding constraint. It does not outperform plain LSB in image quality, and it is not a substitute for a modern block cipher. Its value lies in removing the key-distribution problem that otherwise makes time-derived keys impractical, and in doing so with a mechanism that is independent of the cipher it carries.

6.2. Future Work

1. **Key-derived embedding permutation.** Distributing the payload across the cover under a permutation derived from the pre-shared key would remove the spatial clustering identified in Section 5.3 without reducing capacity, and would deny a block-wise detector its foothold.
2. **Substitution of a standard cipher.** Replacing the substitution cipher with a lightweight block cipher such as SPECK or SIMON [3] would address the known-plaintext vulnerability while retaining the key transport mechanism. Comparing the resulting cost against the 0.102 ms currently attributable to encryption would quantify the trade-off.
3. **Evaluation against modern steganalysis.** Testing against feature-based detectors [15] and against learning-based steganalysis [4] would establish whether the undetectability observed under chi-square and RS analysis holds against current methods.
4. **Robustness to image processing.** Evaluating the effect of lossy compression, cropping, scaling and filtering would determine whether the scheme can be applied outside strictly lossless channels, and would indicate whether error-correcting coding of the payload is warranted.
5. **Header protection.** Investigating whether the header can be protected without introducing a second key-distribution problem would close the known-plaintext foothold described in Section 5.2.
6. **Evaluation on covers with structured histograms.** The chi-square attack is uninformative on highly textured covers such as Baboon. Repeating the evaluation on covers whose byte-value histograms are strongly non-uniform would test the scheme where that attack is actually sensitive.

Acknowledgement

All authors contributed equally to the paper. AI tools were used to translate from Indonesian to English and to check the logical flow of the text. All experimental results, figures and tables were produced by the authors' own implementation, and all numerical values reported were verified against that implementation.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data and Code Availability

The implementation and the notebook that produced every figure and table in this paper are available at https://github.com/rifqipasani-dotcom/timestamp-keyed-lsb-steganography. The cover images are from. The cover images are from the USC-SIPI database [19].

REFERENCES

1. S. N. Al-Rekaby and others, *A Hybrid Security System for Text Encryption and Steganography in Video Using Multi-Level Chaotic Maps*, International Journal of Safety and Security Engineering, vol. 15, no. 3, pp. 521–532, March 2025, DOI: 10.18280/ijsse.150311.
2. S. Amine, F. Koulouh, M. Es-sabry, and N. El Akkad, *Fast and Secure Color Image Cryptosystem Based on 2D Henon Chaotic Map and Josephus-ZigZag Permutation*, Statistics, Optimization and Information Computing, vol. 15, no. 6, pp. 5284–5312, 2026, DOI: 10.19139/soic-2310-5070-3210.
3. R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, *The SIMON and SPECK Lightweight Block Ciphers*, in Proceedings of the 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), pp. 1–6, 2015, DOI: 10.1145/2744769.2747946.
4. M. Boroumand, M. Chen, and J. Fridrich, *Deep Residual Network for Steganalysis of Digital Images*, IEEE Transactions on Information Forensics and Security, vol. 14, no. 5, pp. 1181–1193, May 2019, DOI: 10.1109/TIFS.2018.2871749.
5. C.-K. Chan and L. M. Cheng, *Hiding Data in Images by Simple LSB Substitution*, Pattern Recognition, vol. 37, no. 3, pp. 469–474, March 2004, DOI: 10.1016/j.patcog.2003.08.007.
6. Dafik, S. Venkatraman, G. Sathyanarayanan, R. I. Baihaki, I. L. Mursyidah, and I. H. Agustin, *Enhancing Text Encryption and Secret Document Watermarking through Hyperladder Graph-Based Keystream Construction on Asymmetric Cryptography Technology*, Statistics, Optimization and Information Computing, vol. 14, no. 1, pp. 247–263, 2025, DOI: 10.19139/soic-2310-5070-2310.
7. M. Fayaz and others, *Enhanced PVD Steganography with Dynamic Programming and Rubik's Cube Scrambling for Robust Security*, SSRN Electronic Journal, November 2024, DOI: 10.2139/ssrn.5026044. Preprint.
8. J. Fridrich, M. Goljan, and R. Du, *Detecting LSB Steganography in Color and Gray-Scale Images*, IEEE MultiMedia, vol. 8, no. 4, pp. 22–28, October 2001, DOI: 10.1109/93.959097.
9. W. F. Friedman, *The Index of Coincidence and Its Applications in Cryptanalysis*, Riverbank Laboratories, Geneva, Illinois, 1922.
10. A. D. Ker, *Steganalysis of LSB Matching in Grayscale Images*, IEEE Signal Processing Letters, vol. 12, no. 6, pp. 441–444, June 2005, DOI: 10.1109/LSP.2005.847889.
11. H. Mansouri and others, *Securing Color Images with an Innovative Hybrid Method Combining DNA Computing and Chaotic Systems*, Statistics, Optimization and Information Computing, vol. 12, no. 1, pp. 697–712, May 2024, DOI: 10.19139/soic-2310-5070-1952.
12. J. Mielikainen, *LSB Matching Revisited*, IEEE Signal Processing Letters, vol. 13, no. 5, pp. 285–287, May 2006, DOI: 10.1109/LSP.2006.870357.
13. D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen, *HOTP: An HMAC-Based One-Time Password Algorithm*, RFC 4226, Internet Engineering Task Force, December 2005, DOI: 10.17487/RFC4226.
14. D. M'Raihi, S. Machani, M. Pei, and J. Rydell, *TOTP: Time-Based One-Time Password Algorithm*, RFC 6238, Internet Engineering Task Force, May 2011, DOI: 10.17487/RFC6238.
15. T. Pevný, P. Bas, and J. Fridrich, *Steganalysis by Subtractive Pixel Adjacency Matrix*, IEEE Transactions on Information Forensics and Security, vol. 5, no. 2, pp. 215–224, June 2010, DOI: 10.1109/TIFS.2010.2045842.
16. N. Provos and P. Honeyman, *Hide and Seek: An Introduction to Steganography*, IEEE Security & Privacy, vol. 1, no. 3, pp. 32–44, May 2003, DOI: 10.1109/MSECP.2003.1203220.
17. M. Rifqi and others, *Enhanced Caesar Cipher Algorithm Using Dynamic Key Generation with Timestamp Technique for Data Security*, Journal of Information System and Technology Management, vol. 11, no. 42, pp. 150–162, March 2026, DOI: 10.35631/JISTM.1142009.
18. A. Turakulova and others, *Implementation and Performance Analysis of PVD Method in Image Steganography with XOR-VLSB Enhancement*, International Journal on Informatics for Development, vol. 2025, no. 1, pp. 1–15, June 2025, DOI: 10.14421/ijid.2025.4984.
19. A. G. Weber, *The USC-SIPI Image Database*, Signal and Image Processing Institute, University of Southern California. Available: https://sipi.usc.edu/database/.
20. A. Westfeld and A. Pfitzmann, *Attacks on Steganographic Systems*, in Information Hiding, Lecture Notes in Computer Science, vol. 1768, Springer, pp. 61–76, 2000, DOI: 10.1007/10719724_5.
21. D.-C. Wu and W.-H. Tsai, *A Steganographic Method for Images by Pixel-Value Differencing*, Pattern Recognition Letters, vol. 24, no. 9–10, pp. 1613–1626, June 2003, DOI: 10.1016/S0167-8655(02)00402-6.
22. V. Yadav, M. Hooda, and N. Dhaka, *A Novel Steganography Algorithm based on Interval Type-1 Fuzzy Logic System*, Statistics, Optimization and Information Computing, vol. 14, no. 3, pp. 1512–1524, 2025, DOI: 10.19139/soic-2310-5070-2650.