A Self-Adaptive Bald Eagle Optimized Hybrid Chaotic Framework for Hyperspectral Image Encryption with Reversible Traceability and Merkle-Tree Integrity Verification

Mohd Usaid
Department of Computer Sciences and Engineering, School of Engineering Sciences and Technology,
Jamia Hamdard, New-Delhi-110062
Corresponding author: Mohd Usaid (mohdusaid706@gmail.com)

***Abstract*— Hyperspectral images (HSIs) encode hundreds of contiguous spectral bands per pixel, and adjacent bands routinely share over 95% of their variance. This high inter-band correlation, combined with gigabyte-scale cube sizes, renders conventional ciphers such as AES and RSA computationally impractical and structurally ineffective for HSI protection. Sharma et al. [3] recently showed that a Self-Adaptive Bald Eagle Search (SABES) metaheuristic can be used to autonomously tune chaotic-map parameters for hyperspectral image encryption, removing the need for manual key selection. Building on this SABES-optimized chaotic paradigm, this paper proposes an extended security framework that couples a lightweight three-stage chaotic core—Logistic-map spectral scrambling, 3D Arnold Cat Map spatial permutation, and logistic-keystream XOR diffusion—with SABES-driven optimization of a seven-dimensional parameter set for maximum entropy and minimum correlation, and wraps this core in a dual-layer security shell not addressed by prior SABES-chaos work: Prediction-Error-Labeling (PEL) based Reversible Data Hiding (RDH) for lossless ownership traceability, and a SHA-256 Merkle Hash Tree (MHT) for O(log C) tamper-evident integrity verification. A band-parallel multiprocessing implementation is used to make SABES optimization tractable on high-dimensional cubes. Experiments on a 204-band ENVI-format hyperspectral cube show near-ideal information entropy (7.95–7.99 bits), residual inter-pixel correlation of 0.01–0.03, differential-attack resistance of NPCR ≈ 99.6% and UACI ≈ 33.4%, an effective key space of approximately 2^{325}, and lossless reconstruction (MSE = 0), while an 8-core parallel implementation completes end-to-end encryption of the full cube in 10.95 s, a 7.9× speedup over sequential execution. These results indicate that the proposed extended framework is a practical, real-time-capable security solution for hyperspectral data in remote sensing and defense applications, adding verifiable integrity and traceability to the SABES-chaos encryption paradigm.

Keywords*— *Hyperspectral image encryption; chaotic maps; Logistic map; Arnold Cat map; Self-Adaptive Bald Eagle Search; metaheuristic optimization; Merkle hash tree; reversible data hiding; parallel computing.

I. INTRODUCTION

The rapid growth of remote sensing, defense surveillance, and precision agriculture has made hyperspectral imagery (HSI) a critical data asset. In contrast with traditional RGB pictures composed of three color channels, HSI cubes contain a continuous reflectance spectrum for each pixel over hundred of narrow spectral bands, making material recognition possible in a way that cannot be accomplished via conventional imaging [1]. However, this complexity has a drawback: HSI cubes are characterized by huge sizes and high inter-band correlations, where two neighboring bands exhibit more than 95% of shared variance [2]. If no measures are taken to eliminate these correlations, an enemy could easily perform spectral unmixing and guess the suppressed bands from the leaked ones, making simple encryption methods structurally unsafe. Traditional cryptosystems like AES, DES, RSA, and ECC were invented for text and small binary data payloads. But when applying them to a 3D HSI cube, their intrinsic need for costly per-band implementation, lack of a method to eliminate spectral correlation, and overall ineffectiveness in scalable implementation does not benefit from the advances offered by chaos-based encrypted methods. For instance, chaotic systems depend on deterministic chaos, which is extremely influenced by the initial conditions and enables one to gain pseudo-random output for pixel shuffling and dissemination. Despite the advantages, conventional technologies and

methods still rely on internal techniques since chaotic systems are based on parameters selection and do not allow identifying of data errors in the absence of sender-owner intervention.

According to Sharma et al. [3], the Self-Adaptive Bald Eagle Search (SABES) technique [5] has been shown to efficiently optimize chaotic-map settings for hyperspectral image encryption. The proposed solution, however, talks only about the following aspects of confidentiality: it does not have any methodology for detecting whether the encrypted information has changed its state in any way after encryption or for verifying ownership of the data after it has been sent by the sender. In this paper, we extend the SABES-enhanced method by theorizing its application in two different areas. To start, we establish the chaotic core as a compact three-step confusion-diffusion system optimized for HSI cubes made up of the Logistic-map-based spectral permutation, the 3D Arnold Cat Map spatial permutation, and the XOR encryption diffusion phase, with SABES [5] exploring the seven-dimensional space of parameters to get the configuration that maximizes Shannon entropy and minimizes pixel correlation at the same time with the approach that relies on Band-Parallel Parallelism. Next, we incorporate this chaotic core into a security shell that has not been offered by the preceding SABES-chaos projects, like the PEL-based RDH
[7] for lossless trackability of ownership and the SHA-256 Merkle Hash Tree [8] for quick reasoning. The remainder of this paper reviews related work (Section II), details the proposed framework (Section III), describes the implementation (Section IV), reports experimental results (Section V), discusses their implications (Section VI), and concludes with directions for future work (Section VII).

II. RELATED WORK

Early image encryption research adapted block ciphers such as AES to visual data, but sequential per-band processing of HSI cubes causes severe memory and throughput bottlenecks and does not disrupt spectral cross-correlation, leaving data vulnerable to unmixing attacks [1]. This motivated a shift toward chaos-based confusion-diffusion architectures, first formalized by Fridrich, which offer high parameter sensitivity and natural resistance to differential attacks.

To address HSI-specific redundancy, multi-domain scrambling techniques were proposed by Geng et al. [2], disrupting both spatial and frequency domains; however, such approaches rely on heavy mathematical transforms that add significant computational overhead. Most directly relevant to the present work, Sharma, Singh, and Kaur [3] proposed a hybrid encryption model in which the Self-Adaptive Bald Eagle Search (SABES) algorithm selects the initial and control parameters of a multi-level chaotic system for hyperspectral (including hyperspectral medical) images, and validated it against statistical, differential, noise, and cropping attacks. Their results establish that metaheuristic key search meaningfully outperforms manually tuned chaotic parameters for HSI encryption, and this paper adopts SABES-driven parameter optimization for the same reason. However, [3] addresses confidentiality only and does not provide any mechanism for post-encryption tamper detection or ownership traceability, gaps the present framework is designed to close. Chen, Mao, and Chui [6] separately established the 3D Arnold Cat Map as an effective spatial-permutation primitive for chaotic image encryption, which the present work adopts for per-band spatial scrambling in place of the multi-level chaotic system used in [3].

On the optimization side, Alsattar et al. [5] introduced the Bald Eagle Search (BES) metaheuristic that SABES extends, automatically adjusting search behavior as optimization progresses and improving convergence on complex, multidimensional fitness landscapes such as chaotic-key selection. Separately, robust data stewardship research has emphasized the need for tamper-evidence and provenance in shared or cloud-stored imagery. Hou et al. [4] applied Merkle Hash Tree structures to enable efficient, mathematically sound integrity verification without full-payload inspection, an approach this work extends to per-band hashing of encrypted HSI cubes. For ownership traceability without permanent pixel distortion, Li et al. [7] proposed Prediction-Error-Labeling based reversible data hiding, building on histogram-shifting RDH principles, which the present framework uses to embed and losslessly recover ownership metadata.

Positioned against this background, the present work's contribution is not the use of SABES for chaotic-key optimization in HSI encryption, which [3] already establishes, but the integration of that SABES-optimized chaotic

core with band-parallel execution, PEL-based reversible traceability, and Merkle-tree integrity verification into a single, cohesive pipeline evaluated end-to-end on a real HSI cube — closing the tamper-detection and provenance gaps that confidentiality-only chaotic models such as [3] leave open.

III. PROPOSED FRAMEWORK

A. System Overview

The proposed framework, illustrated conceptually in Fig. 1 (flowchart), consists of six coupled components applied sequentially to a hyperspectral cube: (i) acquisition and pre-processing, (ii) PEL-based reversible watermark embedding, (iii) Logistic-map spectral scrambling, (iv) 3D Arnold Cat Map spatial scrambling,
(v) logistic-keystream XOR diffusion, and (vi) Merkle Hash Tree integrity sealing. Components (iii)–(v) form the chaotic encryption core, whose seven control parameters are jointly optimized by SABES prior to the final encryption pass.

B. Acquisition and Pre-Processing

HSI cubes are loaded in ENVI format (.hdr metadata and .dat pixel data) using the Python spectral library, yielding a Height × Width × Bands array. As the diffusion stage is based on bitwise XOR of the binary numbers, floating point reflectance data has to be normalized and converted to uint8. Given that the Arnold Cat Map needs the size of the spatial matrix to be square, non-square cubes are first padded with zeros before scrambling, and this padding is removed after the decryption process.

C. Reversible Traceability via Prediction Error Labeling

The PEL-based RDH method already embeds a signature that identifies a person or institution before scrambling takes place(2). A Median-Edge Detector creates a model for each pixel with the predictions made based on the values of three pixels: the one directly above, the one to the left, and the diagonal pixel on the upper left. The prediction errors must form a histogram with a peak in its middle and thus a lot of data around zero in order to allow for the shift needed to embed the watermark. Because this shift is totally reversible, by decrypting the message it is possible to recover both the original image and the watermark embedded in it without any distortions.

D. Spectral Scrambling via Logistic Chaotic Map

The Logistic Map, which is described by the formula $X(n+1) = r \cdot X(n) \cdot (1 - X(n))$, $0 < X(n) < 1$, $0 < r \leq 4$, is used to create a chaotic sequence having a length equal to the number of spectral bands of an image. If this sequence is rearranged, a permutation index is obtained, which is then used on the band axis in order to eliminate strong correlation between the different bands, thus rendering all spectral unmixing attack ineffective.

E. Spatial Scrambling via 3D Arnold Cat Map

Every spectral band can be independently modified through the Arnold Cat transformation $x' = (x + a \cdot y) \bmod N$ and $y' = (b \cdot x + (ab+1) \cdot y) \bmod N$ for certain iterations. This process will shuffle the pixels of every band but, irrespective of the number of iterations and parameters a and b used, it can easily be reversed using the same transformation with the same settings.

F. Chaotic Diffusion via XOR Keystream

A second logistic sequence, which has independently been seeded, is developed in 8-bit numbers and molded into a cube of the same size as the scrambled image, thus generating the keystream K. The last voxel that has been scrambled is released into the space using the formula $C(i) = S(i) \oplus K(i)$. As the XOR operator has self-inverse properties, the

same keystream can be applied during the decryption procedure in order to obtain S again. The process ensures uniformity of the histogram of the ciphertext and minimizes inter-pixel correlation to produce the avalanche effect with respect to the message and key in use.

G. SABES-Based Parameter Optimization

Selecting the seven chaotic-core parameters – which are the logistic seeds and control values used for scrambling and diffusion, together with the Arnold constants a, b, and the number of iterations – manually can lead to low randomness and correlation. In this regard, the framework builds upon the SABES-based key-selection method created for hyper-spectral chaotic encryption by Sharma et al. [3]. The implementation of the Self-Adaptive Bald Eagle Search algorithm [5] in the proposed framework ensures the appropriate key-selection procedure known as the Logistic-Arnold-XOR approach as presented in sections III.D–III.F. SABES considers each parameter vector to be a candidate search agent and implements its operations through three phases, which include: (1) Selection phase, characterized by wide-ranging exploration of search space to maintain diversity; (2) Search phase defined by targeted refinement concerning the promising areas of the search space involving neighbourhood dynamics; and (3) Swooping phase,aimed at global optimum through exploitation.
. Each candidate is scored with a fitness function combining the Shannon entropy and horizontal correlation coefficient measured on a representative encrypted band, with invalid evaluations mapped to zero fitness. A greedy update rule retains a candidate only if it strictly improves fitness, and the process repeats until convergence or a fixed iteration budget is exhausted, after which the globally best parameter vector is locked in as the encryption key.

H. Parallel Fitness Evaluation

Each fitness evaluation requires a full encryption pass, making sequential SABES optimization prohibitively slow for large cubes. The framework therefore distributes candidate evaluations across available CPU cores using Python's multiprocessing.Pool, with the unencrypted cube loaded once and shared across workers to minimize memory overhead. Candidate parameter vectors are mapped to workers each iteration, each of which encrypts the image, computes entropy and correlation on a test band, and returns its fitness score for comparison.

I. Integrity Verification via Merkle Hash Tree

After the final encryption pass, each of the C encrypted spectral bands is hashed independently using SHA-256 to form the leaf nodes of a Merkle Hash Tree [4], [8]. Leaf hashes are recursively paired, concatenated, and re-hashed until a single Merkle Root remains, which is stored alongside the ciphertext. Because any bit-level change in any band propagates upward and alters the root, integrity can be verified in O(log C) hash operations by recomputing the root and comparing it against the stored value, without inspecting the payload itself.

J. Decryption

The reverse procedure is followed in decrypting. The diffusion stream of keystream is reconstructed from the parameters fed into the logistic equations. The ciphertext is XORed with the diffusion keystream using the inverse Arnold Cat transformation which applies the stored (a, b) parameters as well as the iteration number to re-establish the spatial structure. The order of the bands is restored using the inverse spectral permuter without any padding used in pre-processing. Since all operations are completely reversible, the cube obtained through this process is identical to the cube that was delivered for encryption.

IV. IMPLEMENTATION

The framework was built using Python 3.10. NumPy provides capabilities for vectorized array manipulation and is used to perform scrambling, XOR diffusion, and correlation calculation; the spectral library processes hyperspectral cubes in the ENVI format (.hdr/.dat); OpenCV implements the histogram calculation needed for entropy and correlation calculations; hashlib provides the SHA-256 hashing for Merkle tree construction; and Numba's @jit decorator helps compile the function for generating the keystream required for logistic mapping into the native code, making the chaotic stream generation much faster. The multiprocessing module in Python, with os.cpu_count(), automatically determines the number of the workers nodes for SABES, which is equal to the number of cores present on the machine

being used (in our case, eight in our environment).

An image was saved in an array format of Height × Width × Bands (in our case, 512 × 512 × 204). The image was converted using uint8 and padded into a square area. The scrambling of the spectral data was done using a permutation procedure based on the sorting of the logistic-map sequence and fancy indexing. The spatial scrambling was done through the application of the Arnold transform using the grid coordinates obtained from the np.meshgrid() procedure. The diffusion was performed using a bitwise xor operation between the scrambled cube and the logistic keystream. SAPES optimization performed five generations of the candidate floor population using Pool.map(), while functools.lru_cache avoided repeated calculations of the fitness function for those parameters repeated in the calculations. Finally, all optimized parameters were used in one confirmation encryption process and the key parameters for logistic map, constants for the Arnold transformation, and data for the spectral scrambling were stored in the JSON key file in encrypted form.

V. EXPERIMENTAL RESULTS AND SECURITY ANALYSIS

The framework was evaluated on a 512 × 512 × 204 ENVI-format hyperspectral cube. Table I provides the main findings related to security and performance metrics of the proposed framework named SABES.

TABLE I. SUMMARY OF SECURITY AND PERFORMANCE METRICS

Metric	Achieved Value	Theoretical Ideal
Information Entropy (bits)	7.95 – 7.99	8.0
Correlation Coefficient	0.01 – 0.03	0.0
NPCR (%)	≈ 99.6	99.6094
UACI (%)	≈ 33.4	33.4635
Reconstruction Error (MSE)	0	0
Effective Key Space	$\approx 2^{325}$	$> 2^{128}$ (secure threshold)
8-core Encryption Time (s)	10.95	—
Parallel Speedup	≈ 7.9×	—

A. Qualitative Visual Analysis

The visual inspection demonstrates that the application of spectral scrambling results in a different order of bands but the employment of the 3D Arnold Cat Map leads to the disappearance of recognizable spatial structure in each band. Thus, the enciphered bands are almost indistinguishable from the noise and the decrypted image fully coincides with the original one, which is particularly significant for remote and military applications.

B. Statistical Security: Entropy and Correlation

The theoretical level of Shannon entropy for an 8-bit image is 8.0, which corresponds to the uniform distribution of pixels. The model under consideration produces the entropy value of 7.95-7.99 for all evaluated bands. The correlation value between neighboring pixels equal to almost 1.0 in the original image falls down to the range of 0.01-0.03 after the implementation of the method presented indicating that a chaotic core is really able to break the spatial correlation that attackers can take advantage of.

C. Differential Attack Resistance

The process of measuring how resistant the encryption method is to differential attacks has been done using two measures: the Number of Pixels Change Rate (NPCR) and the Unified Average Changing Intensity (UACI), in which

according to the used method, the cipher texts from a single plain text modified with one pixel was used. The framework achieved NPCR ≈ 99.6% and UACI ≈ 33.4%, closely matching the theoretical ideals of 99.6094% and 33.4635% respectively, demonstrating strong avalanche sensitivity to minimal plaintext perturbation.

D. Key Space Analysis

The optimized encryption key comprises seven independent chaotic control parameters. Assuming IEEE-754 double-precision resolution ($\approx 10^{14}$ distinguishable values per parameter), the effective key space is approximately $(10^{14})^7 \approx 10^{98} \approx 2^{325}$, which exceeds the widely accepted 2^{128} brute-force security threshold by a large margin.

E. Integrity and Traceability Verification

Recomputing the Merkle root from the retrieved encrypted cube exactly matched the originally stored root, confirming zero undetected tampering across all 204 bands. Independently, the PEL-based watermark extraction recovered the embedded ownership identifier with no residual distortion to the underlying spectral data, validating the lossless-reversibility property central to the traceability layer.

F. Computational Performance

Distributing the 204-band cube's SABES fitness evaluations and final encryption across eight CPU cores reduced total encryption time to 10.95 s, versus a substantially longer sequential baseline, yielding an approximate 7.9× speedup. This confirms that the added optimization and verification layers do not preclude near-real-time operation on commodity multi-core hardware.

VI. DISCUSSION

The results indicate that the three design goals set out in Section I are jointly achievable without significant trade-offs. The use of the SABES method for parameter choice has helped to overcome the historical drawback of chaos encryption technology, which has always relied on the human selection of the potentially non-optimal secret parameters for the function. When measuring data entropy and correlation values, the effectiveness of the parameter optimization mechanism can be confirmed as it obtains the value which is very close to the ideal solution in terms of randomness. At the same time, the implementation of the PEL layer and Merkle tree is illustrated to be compatible with chaotic implementation without imposing any additional burden on the efficiency of the process as obtaining the information takes logarithmically growing time based on the number of utilized bands.

In this case, a constraint can be related to how the fitness function measures the amount of entropy and correlation based on a representative band for each candidate; although this method reduces the costs of the optimization process, it might not be enough to measure the variations in the level of security across different bands. Further, the aforementioned key space data assumes that the only way through which key entropy is calculated is based on using floating point precision; thus, applying a more pessimistic approach, in which practical limitations due to implementation processes are taken into account (for example, violence of chaos caused by limited precision), could make the security claim stronger.

VII. CONCLUSION AND FUTURE WORK

In this article, a hybrid security framework has been presented for hyperspectral images. This framework combines the three-stage chaotic core that is lightweight, along with the autonomous optimization of parameters through the use of SABES and the reversible traceability of ownership through PEL. The integrity of the images is verified by the Merkle Hash Tree, and this entire process is carried out through the parallel processing of data. The HSI cubes having 204 bands have been tested over this hybrid security framework, and the results are near-perfect in terms of entropy (7.95-7.99 bits), almost no correlation (0.01-0.03), as well as good resistance against differential attacks (the NPCR is approx 99.6% and UACI is about 33.4%). There is also a space of keys of about 2^{325}. It can be concluded that this framework can be used to protect huge hyperspectral data in applications such as defense/security.

Some examples of the future research include exploring how hardware acceleration using FPGA technologies can support the chaos core and the fitness analysis of the SABES algorithm in real-time encryption for live satellite or drone videos; studying how to keep Merkle tree roots and water mark extraction data decentralized using a block chain solution; extending the 3D scrambler-diffuser architecture to other 3D imaging methods such as MRI and CT scans offering the same benefits in medical applications.

REFERENCES

[1] C. J. Abbas, M. A. Ibrahim, and M. A. Khan, "An efficient image encryption approach for high-resolution satellite images based on chaotic maps and Blowfish algorithm," Multimedia Tools and Applications, vol. 80, no. 1, pp. 1245–1264, 2021.

[2] W. Geng, J. Zhang, et al., "A hybrid-domain encryption method for hyperspectral remote-sensing images," IEEE Transactions on Geoscience and Remote Sensing, vol. 59, no. 8, pp. 6841–6855, 2021.

[3] S. R. Sharma, B. Singh, and M. Kaur, "A hybrid encryption model for the hyperspectral images: application to hyperspectral medical images," Multimedia Tools and Applications, vol. 83, no. 4, pp. 11717–11743, 2024.

[4] Y. Hou, et al., "Secure content-based image retrieval and traceability in cloud environments using SHSR and CNN extraction," IEEE Transactions on Cloud Computing, vol. 12, no. 2, pp. 450–465, 2024.

[5] H. A. Alsattar, A. A. Zaidan, and B. B. Zaidan, "Novel meta-heuristic bald eagle search optimisation algorithm," Artificial Intelligence Review, vol. 53, no. 3, pp. 2237–2264, 2020.

[6] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," Chaos, Solitons & Fractals, vol. 21, no. 3, pp. 749–761, 2004.

[7] X. Li, B. Yang, and T. Zeng, "Efficient reversible data hiding in encrypted images via prediction error labeling," IEEE Transactions on Information Forensics and Security, vol. 18, pp. 1205–1218, 2023.

[8] R. C. Merkle, "A digital signature based on a conventional encryption function," in Advances in Cryptology — CRYPTO '87, Springer, 1988, pp. 369–378.

[9] C. E. Shannon, "Communication theory of secrecy systems," The Bell System Technical Journal, vol. 28, no. 4, pp. 656–715, 1949.

[10] Y. Wu, Y. Xiang, and Y. Guo, "Reversible data hiding in high-dimensional remote sensing images," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 15, pp. 3412–3425, 2022.