Controller of Certifying Authorities
Ministry of Electronics & Information Technology
Government of India

IEEE COMPUTER SOCIETY Bangalore Chapter

IAS IEEE INDUSTRY APPLICATIONS SOCIETY Linking Research to Practice Bangalore Chapter

5th INTERNATIONAL CONFERENCE ON

PUBLIC KEY INFRASTRUCTURE AND ITS APPLICATIONS (PKIA 2024)

SEPTEMBER 5-6th, 2024

COMPARATIVE ANALYSIS OF HYBRID CRYPTOSYSTEMS FOR SECURE IMAGE ENCRYPTION

NIDHI BHATT, BTECH CSE, INDIRA GANDHI DELHI TECHNICAL UNIVERSITY FOR WOMEN

सी डैक CDAC

https://pkiindia.in

Social Media /pkiindia

IEEE BANGALORE SECTION

INTRODUCTION

Background: Safeguarding sensitive images is vital, especially within Public Key Infrastructure (PKI) systems.

Satellite images are more complex and cover larger areas, requiring more advanced processing compared to normal images. Satellite image encryption is crucial to protect sensitive data from unauthorized access or manipulation.

Traditional text encryption methods (e.g., DES, Triple DES) are not well-suited for images due to their large size.

Objective: This study compares two hybrid cryptosystems for satellite image encryption:

- AES-256-GCM-SHA384
- ChaCha20-Poly1305-SHA256

Both systems use **ECDH** for key exchange and respective algorithms for **bulk encryption**, similar to TLS 1.3. They are evaluated using metrics: **Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), entropy, total encryption/decryption process time, Number of Pixel Change Rate (NPCR), Unified Average Changing Intensity (UACI), correlation coefficients, and Bit Error Rate (BER).**

LITERATURE REVIEW

Authors	Encryption Method	Key Features/Findings
Gerhana et al. [1]	Vigenere cipher	Adapted for images
Putrie et al. [2]	Hill cipher + column transposition	Improved security (PSNR, MSE, SSIM, entropy, histogram)
Ginting et al. [3]	RC4 cipher + chaotic logistic maps	Effective visual encryption
Jolfaei and Mirghadri [4]	Salsa20	Resistant to statistical attacks, limited sensitivity to plaintext changes
Toughi [5]	ECC + AES	Enhanced security with NIST's Elliptic curve random generator
Oktivasari et al. [6]	ECDH + AES-GCM	For ECG image encryption
Chen [7]	AES	Introduced Gini impurity-based factor for quantum threat evaluation
Muhammed et al. [8]	Comparison of 5 symmetric algorithms	ChaCha20 most efficient
Parida et al. [9]	ECC	Resistant to Chosen-Plaintext and Known-Plaintext attacks
Kumar and Sharma [10]	Arnold's cat map + ECC + genetic algorithms	High entropy, low correlation
Mahdi et al. [11]	ChaCha cipher + Hyperchaotic Map	Lightweight, enhanced security, resistant to brute force and statistical attacks

Research gaps addressed in the paper :

1. There has been very little work presented in satellite image encryption, as per my survey.

2. Moreover, the direct comparison of AES-GCM and ChaChaPoly with ECDH key exchange in the context of image encryption has not been presented before.

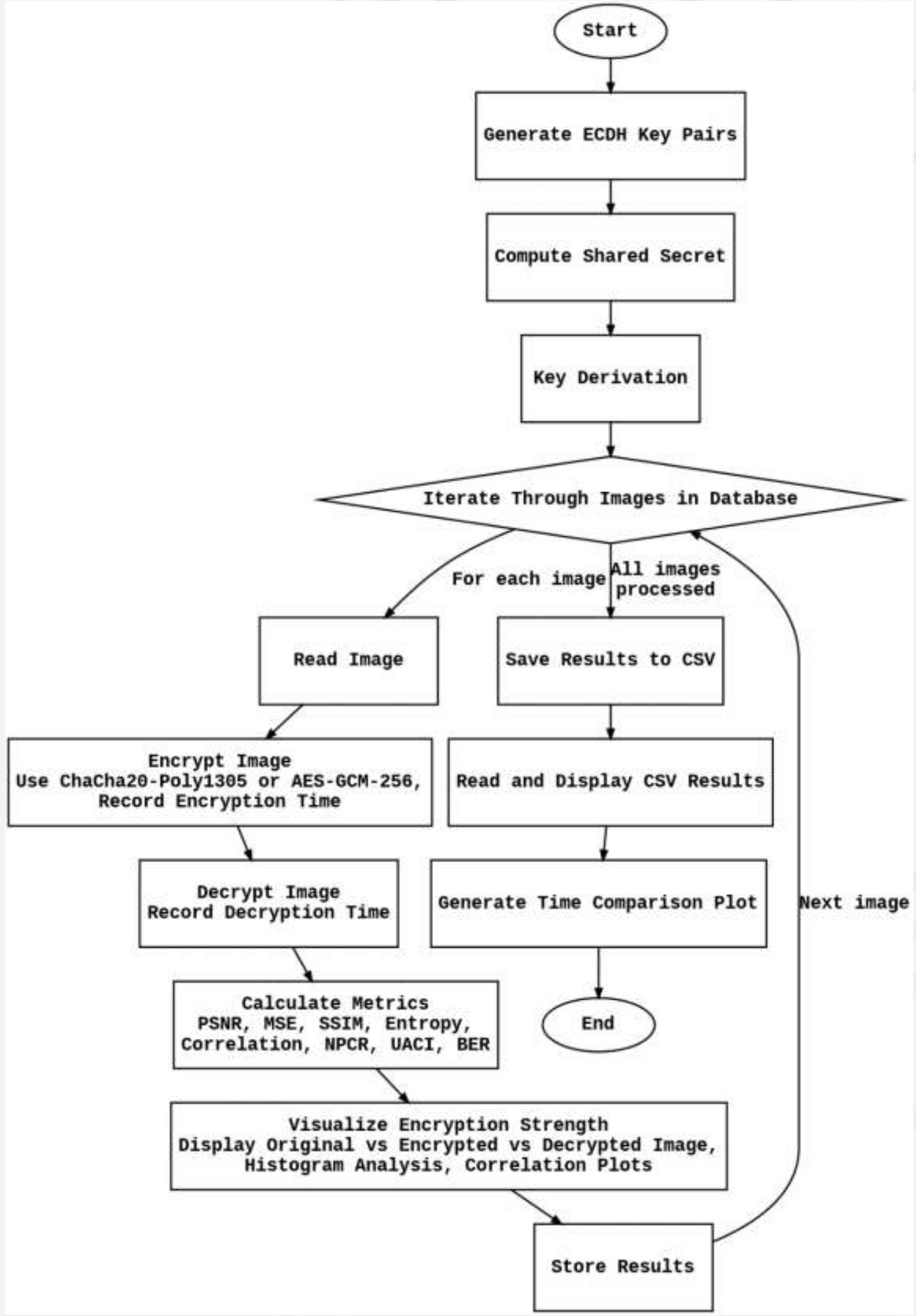
RESEARCH METHODOLOGY

The experiment is done using Python in a **cloud-based Kaggle CPU** environment, which provides a **4-core Intel Xeon processor** running at 2.2GHz with **30GB of RAM**.

The key steps involved are as follows:

1. **ECDH Key Pairs Generation:** Using P-256 curve for 128-bit security.
2. **Shared Secret Computation:** via ECDH algorithm.
3. **Key Derivation:** The shared secret is input into the Hash-based Key Derivation Function **(HKDF)** to derive the final shared key.
4. **Image Database:** 10 satellite images (2448 × 2448 pixels) from DeepGlobe Land Cover Classification Dataset is used for testing. For each image following steps are performed:
   a. **Encryption and Decryption:** Using AES-256-GCM or ChaCha20-Poly1305
   b. **Metrics Calculation:** MSE, PSNR, SSIM, Entropy, Correlation, NPCR, UACI and BER.
   c. **Visualize encryption strength and store results**

5. **Read and display results** for all images and generate a **time comparison plot.**



HYBRID CRYPTOSYSTEMS IMPLEMENTATION

1. Key Exchange:

a) Generate private-public key pairs using **ECDH** using **P-256 curve** (SECP256R1)
b) Exchange public keys
c) Compute shared secret
d) Use HKDF (RFC 5869) to derive final symmetric keys
e) AES-GCM: **SHA-384** for key derivation
f) ChaCha20-Poly1305: **SHA-256** for key derivation

AES-GCM is a block cipher, while **ChaCha20-Poly1305** is a stream cipher.
AES-GCM uses GCM mode, ChaCha20-Poly1305 uses Poly1305 for authentication.
Both are **AEAD (Authenticated Encryption with Associated Data)** algorithms using **256-bit keys.**
.

2. Encryption process:

- Read and flatten image data
- Generate 12-byte nonce using os.urandom()
- Initialize with 32-byte shared key (HKDF-derived)
- Encrypt data with associated data for integrity
- Reshape encrypted data to original image dimensions
- Record encryption time

3. Decryption process:

- Receive encrypted data, nonce, and image shape
- Initialize with same shared key
- Verify authentication tag
- If valid, decrypt data
- Reshape to original image dimensions
- Record decryption time
- Halt if verification fails

PROPOSED IMPLEMENTATION COMPARISON WITH TLS 1.3

1) Similarities:

a. **ECDH Key Exchange:** Uses P-256 curve, similar to TLS 1.3 specifications.

b. **Cipher Suites:** Implements AES-256-GCM and ChaCha20-Poly1305

c. **Key Derivation:** Employs HKDF, consistent with TLS 1.3 practices.

d. **Nonce Generation:** Uses 12-byte nonces.

e. **Authenticated Encryption:** Employ AEAD, a feature also used in TLS 1.3, though with different associated data than what's typically used in TLS 1.3 implementations.

f. **Hash Functions:** Uses SHA-384 and SHA-256 for respective suites.

2) Dissimilarities:

a. **Simplified Handshake:** Lacks full TLS 1.3 handshake process and certificate verification.

b. **Static Keys:** Uses fixed key pairs instead of generating new ephemeral keys for each image.

c. **Specific Use Case:** Tailored for image encryption rather than general-purpose communication.

d. **Simplified Nonce Handling:** Doesn't use TLS 1.3's typical static IV and sequence number combination.

It is important to note that this implementation **does not include a full network stack integration or practical demonstration of HTTPS/SSL.** The focus is on **cryptographic operations** rather than the complete implementation of the TLS 1.3 protocol.

RESULTS AND DISCUSSION

Metric	AES-GCM	ChaChaPoly
MSE	8629.20	8631.19
PSNR (dB)	8.80	8.80
SSIM	0.01	0.01
Entropy	8.00	8.00
KED time (ms)	0.73	1.04
Enc Time (ms)	32.68	39.90
Dec Time (ms)	26.55	28.24
TP Time (ms)	59.95	69.17
NPCR (%)	99.61	99.61
UACI (%)	50.00	49.99
AC	-0.00014	0.00020
BER	0.0	0.0

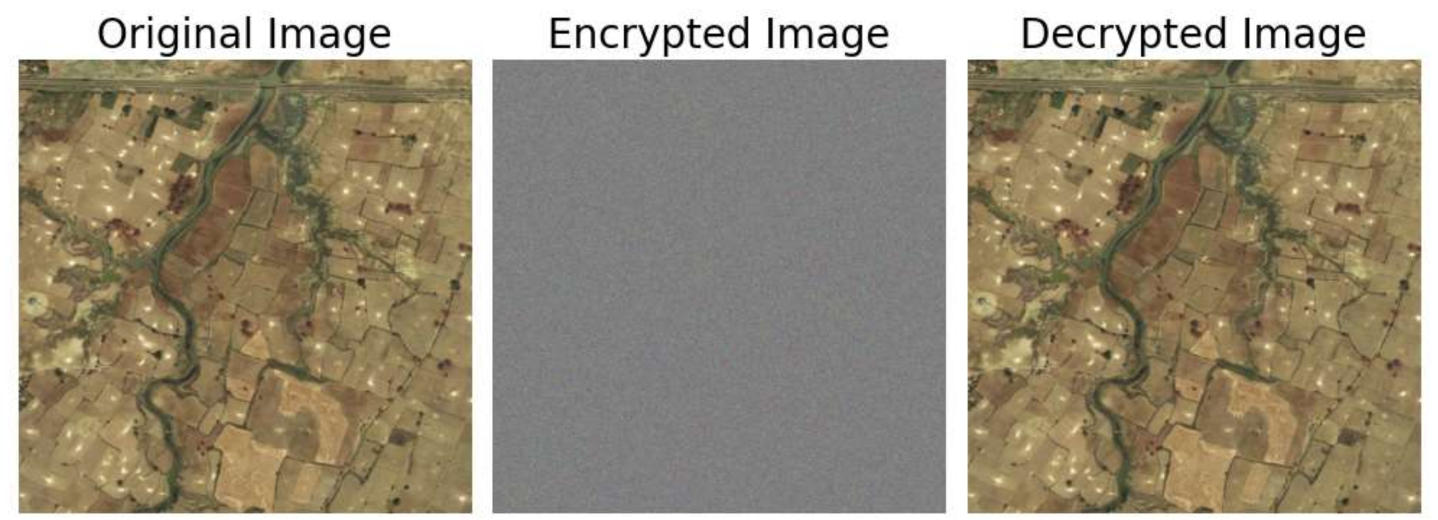
Both systems show similar security performance. The comparison is based on the average of all metrics calculated across the set of images used for testing.

- **AES-256-GCM** faster in all time metrics:

  1. **Key Exchange and Derivation Time:** AES-GCM is **29.81%** faster than ChaChaPoly.
  2. **Encryption Time:** AES-GCM outperforms ChaChaPoly, being **18.09%** faster.
  3. **Decryption Time:** AES-GCM is **5.98%** faster than ChaChaPoly.
  4. **Total Processing Time:** AES-GCM is **13.33%** faster than ChaChaPoly .

- **Near-ideal NPCR** (99.61%) and **UACI** (~50%) values
- Both achieve **maximum entropy** (8.00)
- Average correlation (AC) for encrypted images is near zero, indicating **successful pixel randomization**.

RESULTS AND DISCUSSION CONTINUED....(VISUAL ANALYSIS)



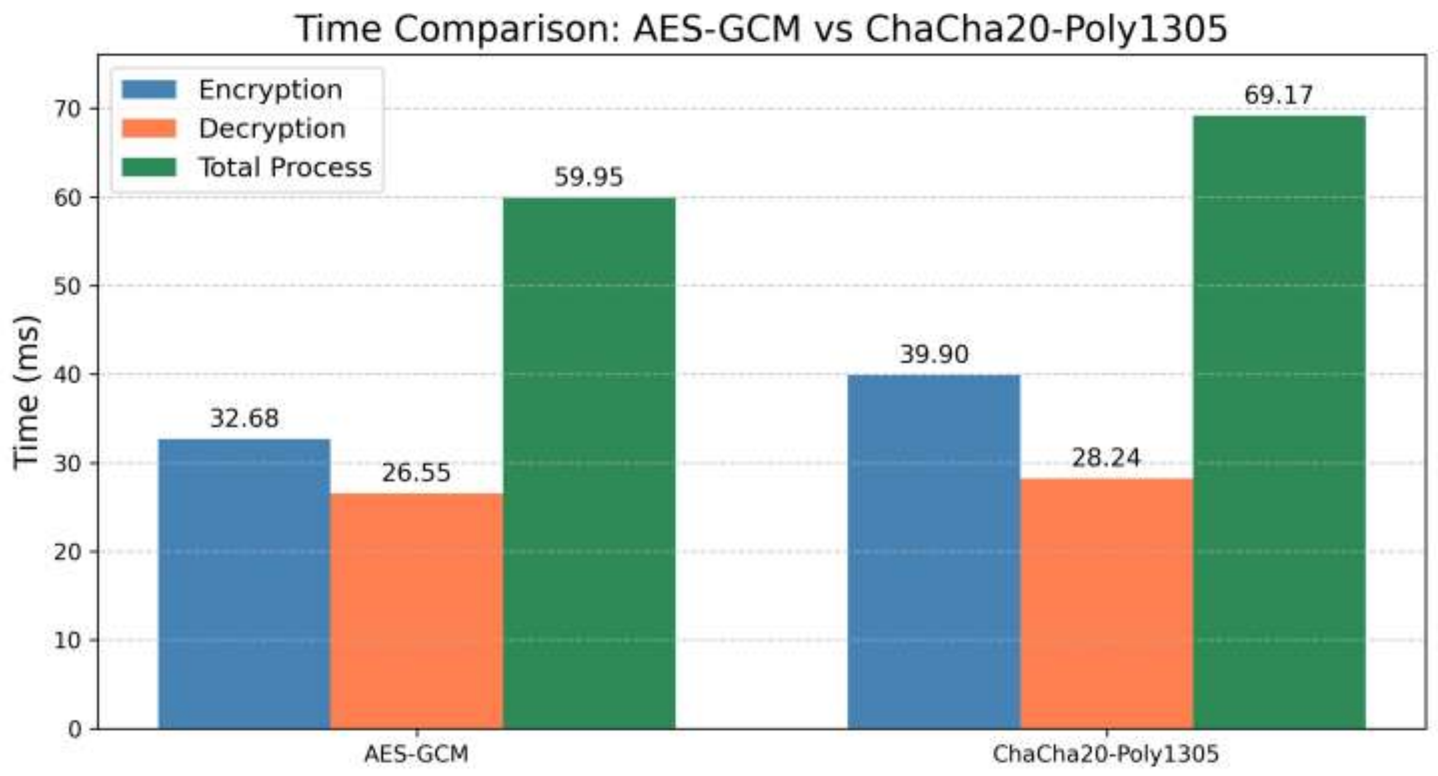


Figure 1

Encrypted image: Indiscernible noise, **no visible patterns**

Decrypted image: Perfect reconstruction, **identical to the original**, BER = 0

This visual comparison demonstrates the strength of the encryption and **the lossless nature** of the process.

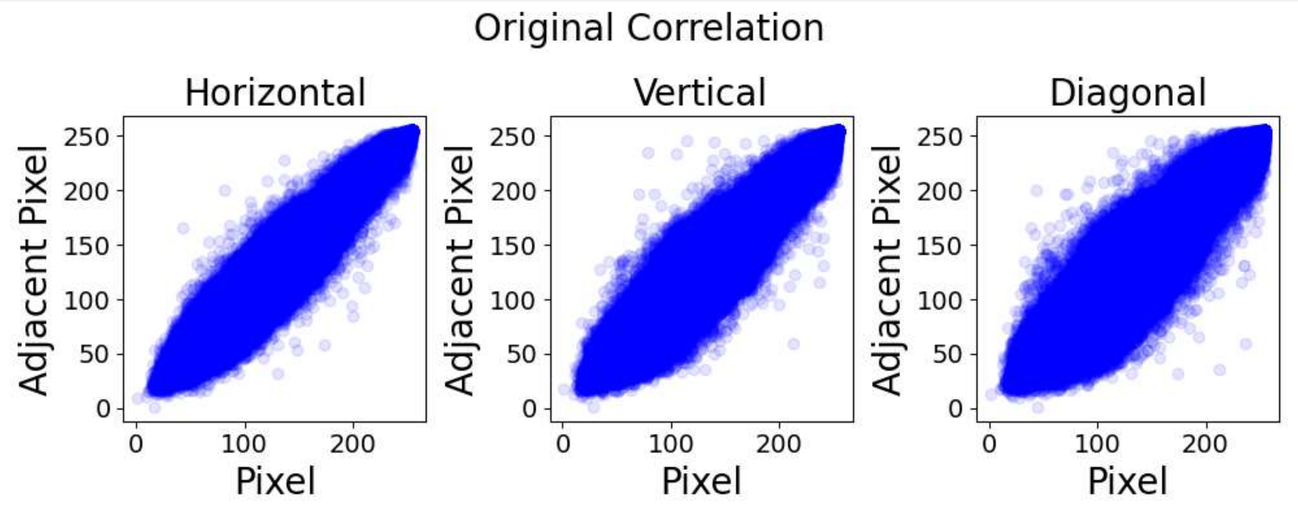
Figure 2

Potential reasons for AES-GCM's superior performance:

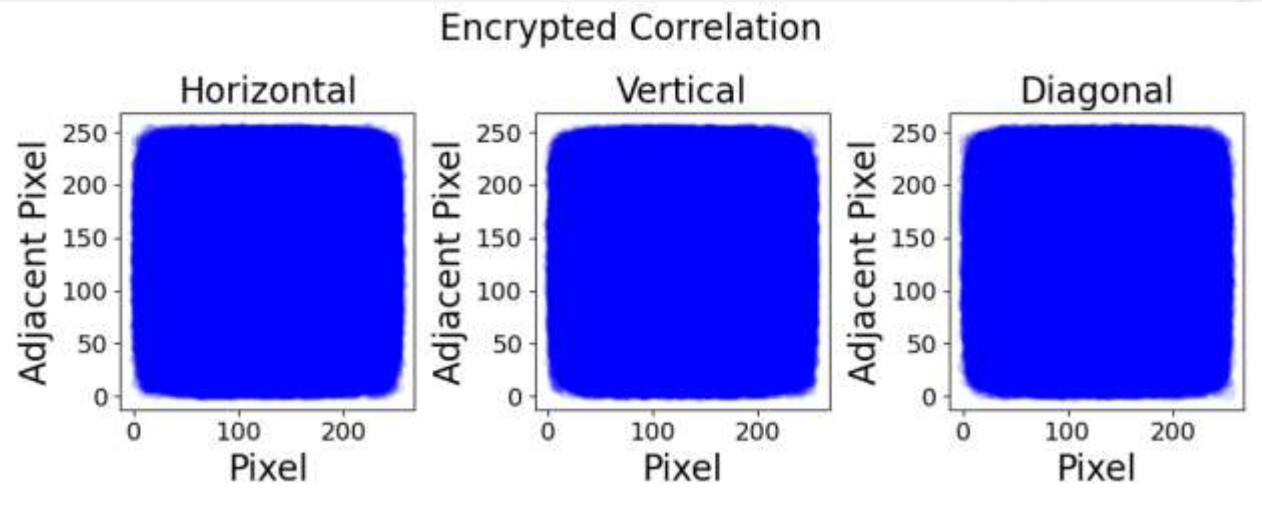
1. **Hardware acceleration for AES** in modern processors.
2. **Implementation environment**

Note: Decryption is faster than encryption due to additional operations in encryption (e.g., **nonce generation**)

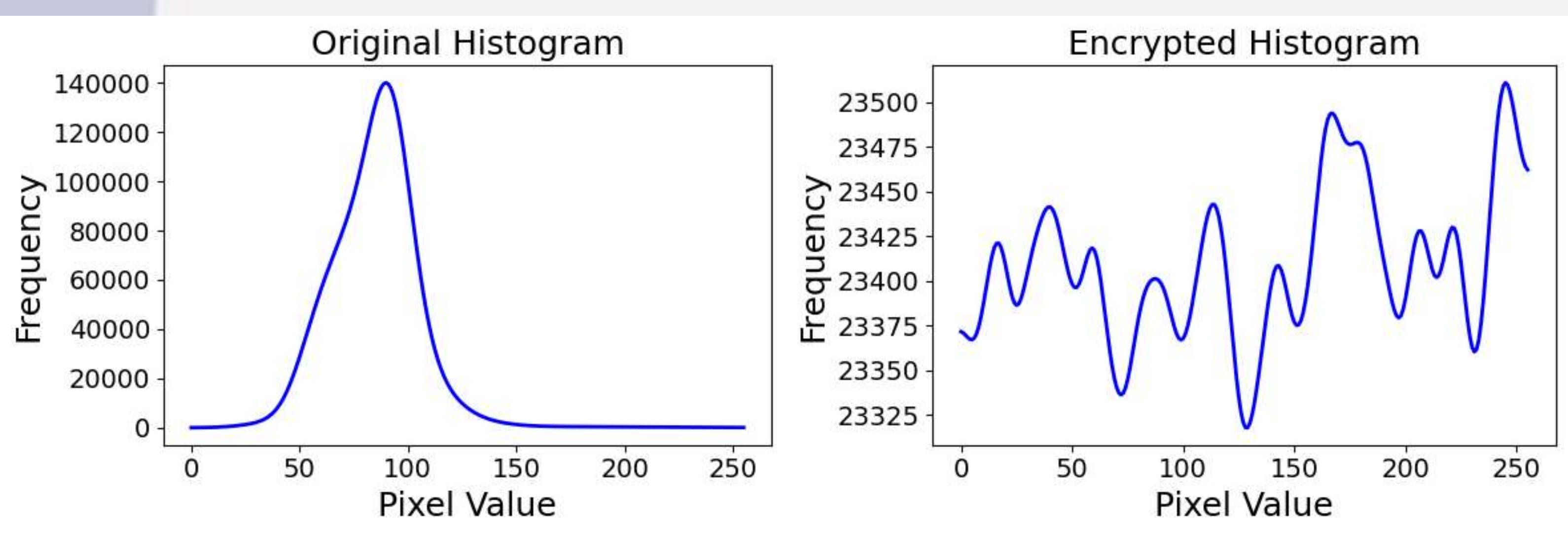
RESULTS AND DISCUSSION CONTINUED….(VISUAL ANALYSIS)



Original image correlation: strong patterns are visible, indicates a high correlation between adjacent pixels.



Encrypted image correlation: uniform distribution, shows successful pixel randomization, near zero correlation coefficients.



Histogram Analysis:

1. **Original histogram:** Non-uniform distribution with distinct peaks.
2. **Encrypted histogram:** Near-uniform distribution across all pixel values.

FUTURE WORK AND CONCLUSION

Proposed future research directions are as follows:

1. **Integration of Post-Quantum Cryptography algorithms**. E.g., CRYSTALS-Kyber, CRYSTALS-Dilithium.
2. Testing with **diverse image types** beyond satellite imagery.
3. Optimization for **resource-constrained environments** (IoT devices) in PKI environments.
4. Advanced security analyses:
    1. **NIST-800-SP-22 tests**
    2. **Gray Level Co-occurrence Matrix Analysis**
5. Comparative studies with **OpenSSL and wolfSSL TLS 1.3 suites.**
6. **Full SSL/HTTPS protocol** stack integration and performance testing against **TLS 1.3 latency requirements.**

Finally, research can be concluded as follows:

1. Both systems exhibit robust security features.
    a. High entropy, near-ideal NPCR and UACI
    b. Effective visual and statistical security
2. AES-GCM outperforms ChaChaPoly in speed, approximately 13.33% faster overall.
3. Practical applications of this work:
    a. Enhancing digital image security in PKI systems
    b. Potential use in e-Governance and IoT applications.

REFERENCES

[1] Yana Aditia Gerhana, Entik Insanudin, Undang Syarifudin, and Mohammad Rizal Zulmi. Design of digital image application using vigenere cipher algorithm. In 2016 4th International Conference on Cyber and IT Service Management, pages 1–5. IEEE, 2016.
[2] Vike Maylana Putrie, Christy Atika Sari, Eko Hari Rachmawanto, et al. Super encryption using transposition-hill cipher for digital color image. In 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pages 152–157. IEEE, 2018.
[3] Riah Ukur Ginting and Rocky Yefrenes Dillak. Digital color image encryption using rc4 stream cipher and chaotic logistic map. In 2013 International Conference on Information Technology and Electrical Engineering (ICITEE), pages 101– 105. IEEE, 2013.
[4] Alireza Jolfaei and Abdolrasoul Mirghadri. Survey: image encryption using salsa20. International Journal of Computer Science Issues (IJCSI), 7(5):213, 2010.
[5] Shahryar Toughi, Mohammad H Fathi, and Yoones A Sekhavat. An image encryption scheme based on elliptic curve pseudo random and advanced encryption system. Signal processing, 141:217–227, 2017.
[6] Prihatin Oktivasari, Maria Agustin, Rahmat Esa Mohammad Akbar, Asep Kurniawan, Ayu Rosyida Zain, and Fachroni Arbi Murad. Analysis of ecg image file encryption using ecdh and aes-gcm algorithm. In 2022 7th International Workshop on Big Data and Information Security (IWBIS), pages 75–80. IEEE, 2022.
[7] Abel CH Chen. Evaluation of advanced encryption standard algorithms for image encryption. In 2024 International Conference on Smart Systems for applications in Electrical Sciences (ICSSES), pages 1–6. IEEE, 2024.
[8] Rebwar Khalid Muhammed, Ribwar Rashid Aziz, Alla Ahmad Hassan, Aso Mohammed Aladdin, Shaida Jumaah Saydah, Tarik Ahmed Rashid, and Bryar Ahmad Hassan. Comparative analysis of aes, blowfish, twofish, salsa20, and chacha20 for image encryption. Kurdistan Journal of Applied Research, 9(1):52–65, 2024.
[9] Priyansi Parida, Chittaranjan Pradhan, Xiao-Zhi Gao, Diptendu Sinha Roy, and Rabindra Kumar Barik. Image encryption and authentication with elliptic curve cryptography and multidimensional chaotic maps. IEEE Access, 9:76191–76204, 2021.
[10] Sanjay Kumar and Deepmala Sharma. A chaotic based image encryption scheme using elliptic curve cryptography and genetic algorithm. Artificial Intelligence Review, 57(4):87, 2024.
[11] Mohammed Salih Mahdi, Raghad Azeez, and Nidaa Falih. A proposed lightweight image encryption using chacha with hyperchaotic maps. Periodicals of Engineering and Natural Sciences (PEN), 8:2138–2145, 11 2020.

Controller of Certifying Authorities
Ministry of Electronics & Information Technology
Government of India

IEEE COMPUTER SOCIETY Bangalore Chapter

IAS IEEE INDUSTRY APPLICATIONS SOCIETY Linking Research to Practice Bangalore Chapter

THANK YOU

The author thanks Mr. Prabhat Kumar, Scientist ‘D’, Ministry of Electronics and Information Technology, Government of India, for his valuable guidance and insights.