SecureVision: A Hybrid Approach to Securing Digital Images

Dhara Lakshmi Kusumanchi, Y. Tanvi, V. Srujan, Anaswara Reji, Kumaran U
Department of Computer Science and Engineering
Amrita School of Computing, Bangalore
Amrita Vishwa Vidyapeetham, India
bl.en.4cse22218@bl.students.amrita.edu,bl.en.u4cse22268@bl.students.amrita.edu,
bl.en.u4cse22262@bl.students.amrita.edu, bl.en.u4cse22273@bl.students.amrita.edu, u_kumaran@blr.amrita.edu

***Abstract*—Secure image transmission and storage are essential in the digital age because of the growing risk of data breaches and illegal access. This paper introducesSeccureVision, a hybrid cryptographic framework that uses a multi-layered method to guarantee the secrecy and integrity of digital images. Even for big image files, the suggested solution offers quick and secure encryption by combining the advantages of two contemporary stream ciphers, XChaCha20 and Salsa20. For integrity verification, BLAKE2b-256, a fast cryptographic hash function, is used to prevent manipulation and confirm authenticity. Images are kept private and unchanged throughout their existence thanks to the layered application of hashing and encryption.SecureVision is suitable for real-time and resource-constrained applications because it strikes a good mix between security, performance, and resource efficiency, according to experimental tests. With no technical complexity, the framework's user-friendly implementation allows for safe image encryption, decryption, and hash validation.**

***Index Terms*—Image encryption, stream cipher, XChaCha20, Salsa20, BLAKE2b, data integrity, cryptographic hash, image security, hybrid cryptography, and secure image transmission**

I. Introduction

Sensitive visual data is being transmitted and stored at an exponential rate due to the expansion of digital communication and multimedia content. Images are an essential type of data, from private images posted on social media to key satellite imagery utilized in scientific and defense applications. However, because of their extensive use, they are also a great target for cyberattacks that include distribution, alteration, interception, and unauthorized access. Thus, protecting the integrity and security of picture data has emerged as a key goal in contemporary information systems.The performance, economy, and integrity requirements of huge image files are frequently beyond the scope of conventional cryptographic systems, despite their effectiveness in protecting textual data. Fast yet robust encryption techniques that may preserve data security without sacrificing performance or system resources are necessary for high-resolution photos. Furthermore, a lot of common encryption techniques lack built-in tools for confirming the validity and integrity of data, making encrypted photos susceptible to covert manipulation.

The study suggestsSecureVision, a novel hybrid cryptographic architecture created especially for safe image encryption and integrity verification, as a solution to these issues. The framework integrates the features of two contemporary stream ciphers, XChaCha20 and Salsa20, which are renowned for their excellent performance, scalability, and resilience to cryptographic assaults. With its longer nonce (192 bits), XChaCha20, an extension of the ChaCha20 algorithm, greatly lowers the likelihood of nonce reuse and provides increased security for systems with big data volumes or lengthy runtimes. In contrast, Salsa20 offers quick and lightweight encryption, which makes it perfect for managing huge multimedia files without causing noticeable processing delay.SeccureVision incorporates the BLAKE2b-256 cryptographic hash function to guarantee data integrity and identify illegal changes. BLAKE2b is known for its speed, ease of use, and security; it outperforms more traditional hash functions like MD5 and SHA-1 and provides defense against frequent hash assaults. With a graphical user interface that enables users to encrypt photos, check their integrity, and maintain cryptographic keys without the need for extensive technical knowledge, the system is made to be both user-centric and accessible. Thus,SeccureVision provides a workable solution for actual image protection situations by bridging the gap between strong security and usability.

The paper describes the architecture, techniques, and implementation ofSeccureVision in detail. Through trials, it also assesses the system's security and performance, proving its suitability for use in digital forensics, cloud storage, secure picture transmission, and other fields where image integrity and secrecy are crucial.

II. Literature Survey

The study [11] aims to compare the effectiveness of the Advanced Encryption Standard (AES) and Salsa20 algorithms using Novel Noise Images as encryption keys. AES outperformed Salsa20 in accuracy (93.96% vs 61.95%) with statistical significance ($p = 0.001$), making it more suitable for secure file storage and transfer. G Power analysis with a sample size of 10 supported these findings.

Paul Knutson's thesis [12] delves into the security of the lightweight Salsa20 cipher. Utilizing deep learning via a Context Aggregation Network (CAN) and Hamming distance-based differential analysis, the study reveals a weakness in the quarter-round function's avalanche effect. Nevertheless, the

overall algorithm remains secure and resistant to deep learning attacks.

An innovative video encryption strategy is proposed in [13], where only facial margins are encrypted using the Salsa20 algorithm after detection via the Viola-Jones method. The approach improves processing time (1.033 seconds) and maintains visual quality, suitable for real-time video applications.

The work in [14] explores Salsa20's use in image encryption, emphasizing its lightness and adaptability. It discusses key and nonce usage, and benchmarks Salsa20's speed and robustness against other encryption schemes, highlighting its suitability for real-time image protection.

Another study [15] reinforces Salsa20's strength in image encryption. The research underscores the importance of safeguarding visual data and contrasts Salsa20's lightweight performance against traditional algorithms in real-time settings.

The survey in [16] categorizes image encryption methods into symmetric and asymmetric techniques, stressing evaluation parameters like histogram uniformity and key sensitivity. It includes methods using neural networks, pixel substitution, and permutation, and calls for continued exploration in image security.

Study [17] evaluates end-to-end group chat security in popular messaging apps like Signal, WhatsApp, and Threema. It uncovers vulnerabilities in group message integrity and administration, and highlights the absence of features like future secrecy, suggesting countermeasures for robust communication.

Paper [18] integrates deep learning for secure wireless communication, focusing on user authentication and secret data transmission. It introduces a modified loss function and "symbol-level fingerprints" for robust performance against eavesdropping and varying channel conditions.

Delay Tolerant Networks (DTNs) and mobile ad hoc networks are discussed in [19] for secure communication. A super node-based design uses symmetric and asymmetric encryption to ensure authentication and efficient routing, reducing unauthorized traffic and resource misuse.

To improve cloud backup security, study [20] extends EncFS with user-specific keys and a hierarchical key management system. This structure ensures only authorized decryption and adds minimal overhead to the original file system while improving security.

A real-time group chat system is presented in [21], which ensures end-to-end encryption using a transient key-sharing mechanism without server-side access. It outperforms existing apps like Signal and Telegram for temporary, confidential communication.

Cloud data management is addressed in [22] through "sticky policies" that enforce privacy rules across service providers. The solution uses PKI, IBE, and secret sharing to offer fine-grained control and accountability during data transfers.

The Group Off-the-Record (GOTR) protocol introduced in [23] extends secure messaging to multi-user environments. It enables deniability, confidentiality, and forward secrecy through a "virtual server" model, validated with a GAIM client plugin.

In [24], the challenges of multi-user encrypted cloud databases are tackled using fully homomorphic encryption and blind decryption. This allows collaborative, secure querying without sharing secret keys, even after user revocation.

The EU's Digital Markets Act and its implications for cross-platform E2EE messaging are discussed in [**?**]. The proposed architecture addresses identity resolution, secure key exchanges, and abuse mitigation, pointing out the trade-offs between interoperability and privacy.

Blockchain and IPFS are leveraged in [26] to create a decentralized digital notarization system. With selective disclosure, video-based identity checks, and seven microservices, the model ensures privacy, transparency, and tamper-proof document handling.

The formal security of the Signal protocol is analyzed in [27], modeling its key exchange as a multi-stage AKE with a tree-based structure. The study confirms its features like forward secrecy and post-compromise security through formal proofs.

In [28], the tension between public safety and user privacy in the context of WhatsApp's E2EE is examined. It argues against backdoors, asserting that weakening E2EE undermines human rights without effectively curbing criminal activities.

A hybrid AES-RSA approach to E2EE is evaluated in [29], balancing security and performance. The study recommends better cryptographic choices and user education to enhance adoption and understanding of encryption technologies.

Lastly, [30] proposes a game-theory-based deception strategy to protect communications in cyber-social systems. By misleading attackers through controlled information flow, the method maintains usability while enhancing security.

III. Methodology & Implementation

A. System Design and Architecture

The encryption and decryption technology used in this study is designed to process sensitive data with great security and efficiency. The system consists of multiple layers: the encryption module, decryption module, and the integrity checking module. The encryption module secures plaintext using cryptographic algorithms, while the decryption module restores the original data. The integrity module verifies the data's authenticity during storage or transmission. Input data undergoes a workflow involving key generation, nonce handling, encryption, authentication, and storage. Decryption follows the reverse order, including integrity verification.

B. Algorithmic Flow

Understanding BLAKE2b-256, XChaCha20, and Salsa20 is crucial for secure image encryption and decryption.

1) Workflow for Salsa20 Encryption: Salsa20 is a 512-bit stream cipher that generates a pseudorandom keystream, XORed with plaintext to produce ciphertext. It initializes a 512-bit state array with constants, keys, nonces, and block numbers, then applies 20 rounds of quarter-round transformations involving XOR, rotation, and addition. Finally, the keystream is XORed with image data.(as shown in Fig. 1)

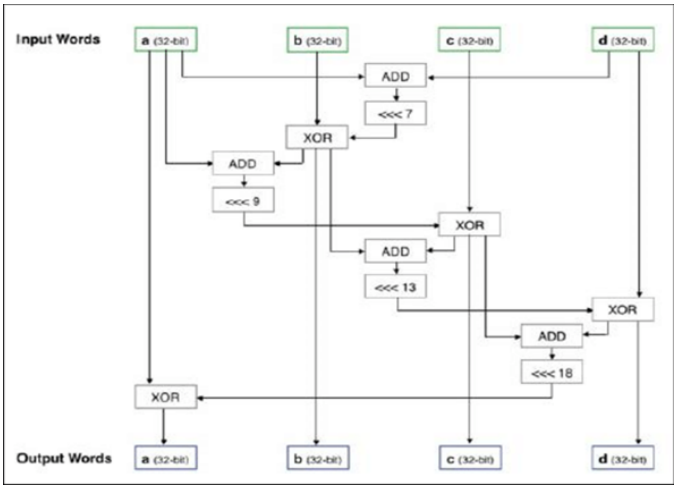

Fig. 1: Workflow of Salsa20 Encryption

2) Workflow for XChaCha20 Encryption: XChaCha20 uses a 192-bit nonce, enhancing resistance to reuse attacks. It first derives a subkey from the 32-byte key and the first 16 bytes of the nonce using HChaCha20. This subkey, along with the remaining 8-byte nonce and block counter, initializes the ChaCha20 cipher. 20 rounds of operations are then applied for encryption.(as shown in Fig.2)

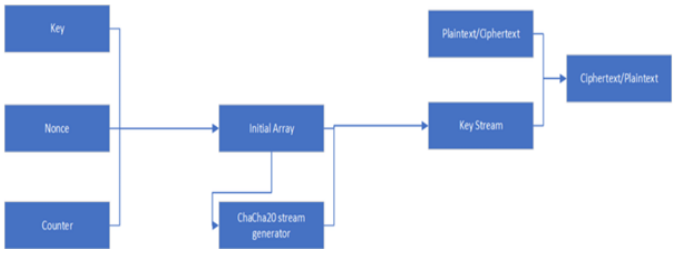

Fig. 2: Workflow of XChaCha20 Encryption

3) Workflow for BLAKE2b-256 Hash Computation: BLAKE2b-256 generates a 256-bit MAC for verifying integrity. It initializes a 512-bit state with constants and a secret key, then processes each message block using G-function transformations over 12 rounds. The final digest is used to check integrity.

C. Implementation of Encryption Algorithm

1) User Input and File Handling: Users upload image files through a web interface. The system checks for file validity, generates dynamic keys and two nonces: 8-byte for Salsa20 and 24-byte for XChaCha20, to ensure unpredictability and protect against replay attacks.

2) Salsa20 and XChaCha20 Encryption: The uploaded image undergoes two-layer encryption: first by Salsa20 using the 8-byte nonce and key, then by XChaCha20 with a 24-byte nonce. Finally, BLAKE2b-256 generates a MAC to ensure integrity.(as shown in Fig.3)

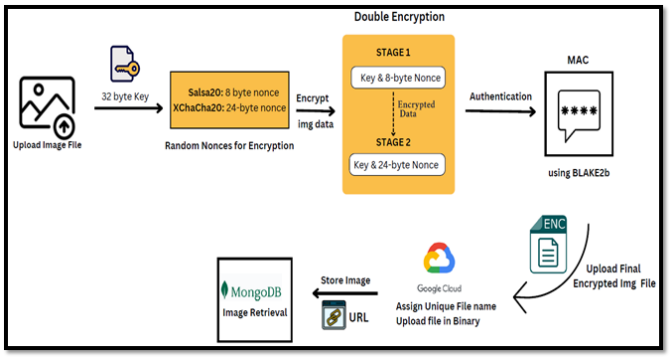

Fig. 3: Combined Salsa20 and XChaCha20 Encryption Process

3) Safe Storage in Google Cloud Storage (GCS): The encrypted image is stored in GCS with controlled access. Encryption keys and nonces are kept separate from the image.

4) Storage of URLs in MongoDB Atlas: Instead of storing the image directly, its GCS URL and metadata (nonces, hashes) are stored in MongoDB Atlas, improving security and separation of concerns.

D. Verification and Decryption of Images

During decryption, the image is downloaded from GCS, then decrypted using XChaCha20 followed by Salsa20. BLAKE2b verifies integrity by comparing hash values. (as shown in Fig.4)

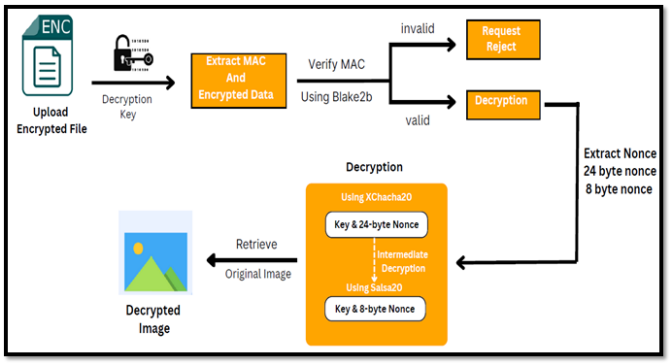

Fig. 4: Image Decryption and Integrity Verification

E. Security Features

The system uses HTTPS, role-based access, API key restrictions, session management, and per-session dynamic key generation to ensure maximum security.

F. Implementation on Render

The platform is hosted on Render using Docker. Flask (Python) powers the backend, while HTML, CSS, and JS handle the frontend. Continuous deployment allows seamless updates. GCS handles encrypted storage, and MongoDB Atlas stores metadata.

IV. Results

1) Encryption of Image: Users can encrypt images via a user-friendly web interface. Options include: Encrypt, Decrypt, and Decrypt from GCS. The output is a '.enc' file. (as shown in Fig. 5)

Fig. 5: Web Interface for Image Encryption

2) Decryption of Image: To decrypt, users upload the '.enc' file and secret key. The platform restores the original image upon successful authentication.(as shown in Fig. 6)

Fig. 6: Web Interface for Image Decryption

3) Encrypted Image File: The encrypted '.enc' image file generated post-encryption is shown in Fig. 7.

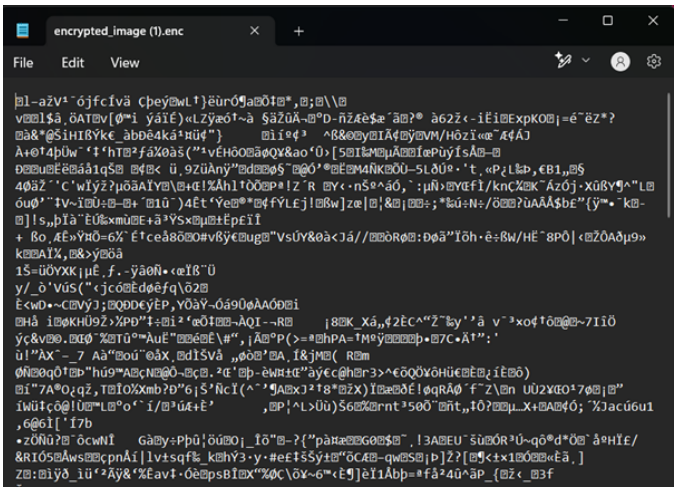

Fig. 7: Encrypted Image File

4) Decrypted Output: Once decrypted using the key, the original image is successfully restored.(as shown in Fig. 8)

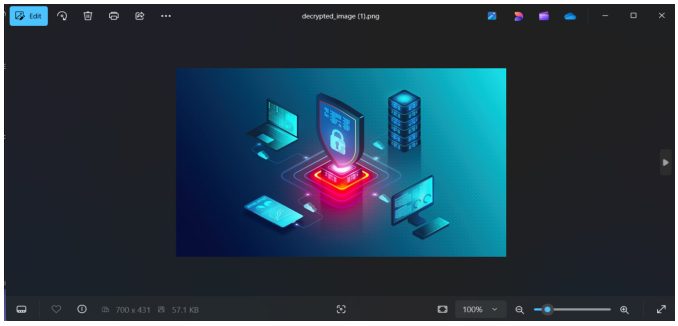

Fig. 8: Output: Decrypted Image

V. Conclusion and Future Scope

SeccureVision offers a strong and effective cryptographic framework for protecting digital images by combining the advantages of the Salsa20 and XChaCha20 stream ciphers into a layered encryption technique. The system successfully detects any unwanted modifications and guarantees data security and integrity with the addition of BLAKE2b-256 hashing. According to experimental results,SeccureVision offers a lightweight and intuitive interface together with high encryption-decryption accuracy, rapid performance, even on high-resolution images, and dependable tamper detection. Because of these characteristics, it is ideal for uses like private cloud storage, forensic data protection, and safe media sharing. In the future, the framework can be improved even more by adding client-side encryption for safe cloud synchronization, cross-platform mobile support, and wider file type compatibility, such as for documents and videos.The system's capabilities might also be further increased by adding features like biometric-based key access, AI-assisted tamper detection, and GPU-accelerated processing, which would makeSeccureVision a scalable and intelligent security solution for the changing needs of digital content protection.

References

[1] Pakshwar, Rinki, Vijay Kumar Trivedi, and Vineet Richhariya. "A survey on different image encryption and decryption techniques." International journal of computer science and information technologies 4.1 (2013): 113-116.

[2] Devi, Aarti, Ankush Sharma, and Anamika Rangra. "A review on DES, AES and blowfish for image encryption decryption." International Journal of Computer Science and Information Technologies 6.3 (2015): 3034-3036.

[3] Saraf, Kundankumar Rameshwar, Vishal Prakash Jagtap, and Amit Kumar Mishra. "Text and image encryption decryption using advanced encryption standard." International Journal of Emerging Trends Technology in Computer Science (IJETTCS) 3.3 (2014): 118-126.

[4] Zhou, Nan-Run, et al. "Novel multiple color images encryption and decryption scheme based on a bit-level extension algorithm." Expert Systems with Applications 238 (2024): 122052.

[5] Muhammed, Rebwar Khalid, et al. "Comparative analysis of aes, blowfish, twofish, salsa20, and chacha20 for image encryption." arXiv preprint arXiv:2407.16274 (2024).

[6] Zhou, Ri-Gui, et al. "Quantum image encryption and decryption algorithms based on quantum image geometric transformations." International Journal of Theoretical Physics 52 (2013): 1802-1817.

[7] Chowdhary, Chiranji Lal, et al. "Analytical study of hybrid techniques for image encryption and decryption." Sensors 20.18 (2020)

[8] Kaur, Manjit, and Vijay Kumar. "A comprehensive review on image encryption techniques." Archives of Computational Methods in Engineering 27.1 (2020)

[9] Gafsi, Mohamed, et al. "Improved chaos-based cryptosystem for medical image encryption and decryption." Scientific Programming 2020.

[10] Noor, Noor Sattar, et al. "A fast text-to-image encryption-decryption algorithm for secure network communication." Computers 11.3 (2022)

[11] Jashwanth, J., and Kalimuddin Mondal. "Improved Accuracy in File Encryption using Noise Images as Key for AES Algorithm in Comparison with Salsa20 Algorithm." 2024 Ninth International Conference on Science Technology Engineering and Mathematics (ICONSTEM). IEEE, 2024

[12] Knutson, Paul. Vulnerability analysis of Salsa20: differential analysis and deep learning analysis of Salsa20. MS thesis. Universitetet i Sørøst-Norge, 2020

[13] Abd Al-Munaf, Ayat Muhammed, Abeer Salim Jamil, and Nidaa Flaih Hassan. "Encryption Edge Points in Face using Light Stream Algorithm." 2022 3rd Information Technology To Enhance e-learning and Other Application (IT-ELA). IEEE, 2022

[14] Jolfaei, Alireza, and Abdolrasoul Mirghadri. "Survey: image encryption using Salsa20." International Journal of Computer Science Issues (IJCSI) 7.5 (2010)

[15] Alghamdi, Yousef, and Arslan Munir. "Image encryption algorithms: a survey of design and evaluation metrics." Journal of Cybersecurity and Privacy 4.1 (2024)

[16] Kumar, Mohit, Akshat Aggarwal, and Ankit Garg. "A review on various digital image encryption techniques and security criteria." International Journal of Computer Applications 96.13 (2014)

[17] Rösler, Paul, Christian Mainka, and Jörg Schwenk. "More is less: On the end-to-end security of group chats in signal, whatsapp, and threema." In IEEE European Symposium on Security and Privacy (EuroSP), pp. 415-429. IEEE

[18] Sun, Zhuo, Hengmiao Wu, Chenglin Zhao, and Gang Yue. "End-to-end learning of secure wireless communications: confidential transmission and authentication." IEEE Wireless Communications 27, no. 5 : 88-95

[19] Samuel, Hany, and Weihua Zhuang. "Preventing Unauthorized Messages and Achieving End-to-End Security in Delay Tolerant Heterogeneous Wireless Networks." J. Commun. 5, no. 2 : 152-163

[20] Leibenger, Dominik, Jonas Fortmann, and Christoph Sorge. "Encfs goes multi-user: Adding access control to an encrypted file system." In IEEE Conference on Communications and Network Security (CNS), pp. 525-533. IEEE

[21] Melo, Tiezer, António Barros, Mário Antunes, and Luís Frazão. "An end-to-end cryptography based real-time chat." In 16th Iberian Conference on Information Systems and Technologies (CISTI), pp. 1-6. IEEE

[22] Beiter, Michael, Marco Casassa Mont, Liqun Chen, and Siani Pearson. "End-to-end policy based encryption techniques for multi-party data management." Computer Standards Interfaces 36, no. 4 : 689-703

[23] Bian, Jiang, Remzi Seker, and Umit Topaloglu. "Off-the-record instant messaging for group conversation." In IEEE International Conference on Information Reuse and Integration, pp. 79-84. IEEE

[24] Gahi, Youssef, and Imane El Alaoui. "A secure multi-user database-as-a-service approach for cloud computing privacy." Procedia Computer Science 160 : 811-818

[25] Len, Julia, Esha Ghosh, Paul Grubbs, and Paul Rösler. "Interoperability in end-to-end encrypted messaging." Cryptology ePrint Archive (2023)

[26] Cosmin-Iulian, Irimia, and Iftene Adrian. "Decentralized Infrastructure for Digital Notarizing, Signing and Sharing Documents Securely using Microservices and Blockchain." IEEE Access (2024)

[27] Cohn-Gordon, Katriel, Cas Cremers, Benjamin Dowling, Luke Garratt, and Douglas Stebila. "A formal security analysis of the signal messaging protocol." Journal of Cryptology 33 (2020): 1914-1983

[28] Endeley, Robert E. "End-to-end encryption in messaging services and national security—case of WhatsApp messenger." Journal of Information Security 9, no. 1 (2017): 95-99

[29] Blaise, Ohwo Onome, Oludele Awodele, and Odunayo Yewande. "An Understanding and Perspectives of End-To-End Encryption." Int. Res. J. Eng. Technol.(IRJET) 8, no. 04 (2021): 1086

[30] Omolara, Abiodun Esther, Aman Jantan, Oludare Isaac Abiodun, Kemi Victoria Dada, Humaira Arshad, and Etuh Emmanuel. "A deception model robust to eavesdropping over communication for social network systems." IEEE Access 7 (2019): 100881-100898