International Journal of Scientific Research in Computer Science, Engineering and Information Technology

ISSN : 2456-3307 OPEN ACCESS

Available Online at : www.ijsrcseit.com

doi : https://doi.org/10.32628/CSEIT25112859

EvidenceX: A Comprehensive Digital Forensics Tool for Evidence Extraction

Nakul Sharma, Sumit Minhas, Karanjit Singh, Vipin, Suman Hait, Lakshyaraj Singh Rathore

School of Computer Application, Lovely Professional University, Phagwara, India

ARTICLEINFO

Article History:

Accepted : 25 April 2025

Published: 27 April 2025

Publication Issue

Volume 11, Issue 2

March-April-2025

Page Number

3828-3836

ABSTRACT

Background: Digital forensics operates as a core investigative element in present times since analysts need to examine metadata and prove file authenticity while locating hidden information. High-end file systems such as XFS and Btrfs include sophisticated features for journaling and copy-on-write that complicate the tasks of data forensic recovery as well as examination processes. Modern forensic tools focus on retrieving information but they demonstrate limited capability in producing extensive metadata analysis together with timeline reports.

Investigative teams gain better results from checking times- tamps along with source device IDs and file historical logs to prove evidence authenticity compared to simply using deleted file recovery methods. File integrity remains secure because cryptographic hash values function as digital identification marks for authentication validation. Supplies of digital content with con- cealed data require identification to retrieve vital incriminating evidence.

The modern forensic requirements have their solution in Evi- denceX which represents a state-of-the-art digital forensic utility. The system performs metadata extraction from multiple file types while automatically finding concealed content and analyzes both system logs and unusual behavior patterns simultaneously with steganography detection capabilities. Anomaly detection functionality in the system protects data consistency while the ”de-forensic” innovation allows secure metadata removal for privacy support. The present document examines how EvidenceX functions and demonstrates its operational value to boost digital forensic procedures through improved accuracy alongside effi- ciency and reliability aspects.

Index Terms —- Digital Forensics, Metadata Extraction, Anomaly Detection, Activity Log Analysis, Privacy Protection.

Introduction

Some investigators responsible for handling digital inves- tigations use their skills to search out and confirm electronic proof for criminal investigations of breaches and cybersecurity cases. Digital forensic tools require immediate development because expanding digital information quantities coincide with advanced cyber threats [1].

EvidenceX Digital Forensic Tool represents a modern all- encompassing solution which specifically meets the technolog- ical requirements of current digital investigative needs. The advanced software platform supports varied advanced func- tions from extracting detailed metadata to detecting concealed files and analyzing steganography patterns and monitoring file changes while recording all activities.

EvidenceX delivers powerful digital investigation tools to investigators which provides both high efficiency and precise examination of digital evidence. Both data visibility and integrity receive enhancement from this system because these factors determine evidence admissibility and reliability in legal proceedings. The ”de-forensic” module operates as a built- in feature of the tool for safe metadata deletion that defends privacy while cleaning up data.

EvidenceX delivers substantial operational improvements to law enforcement personnel and digital forensic analysts and cybersecurity professionals through its advanced analytical methods united with user-oriented interface. The paper inves- tigates the operational advantages of the tool to showcase how its innovative features enhance both accuracy and efficiency as well as reliability in digital forensic investigations.

METHODOLOGY

The development and operation of EvidenceX follow a structured methodology centered around integrating special- ized tools and libraries within a modular Python framework to achieve its primary forensic objectives. The working Process of this tool is explain in figure1

TABLE I

TABLE I

Author	Goal	Technique	Purpose	Limitations
[1]	Handwritten Text Recognition	Access, Accuracy, Cataloging, Guidance, Enhancement	Digitization, Bias, Omission, Ethics, Sustainability	Dependency on digitization pipelines, risk of bias, omission of archival history, legal/ethical issues, and environmental costs
[2]	Integrity, Security, Multi-Cloud Verification, Resilience	Verified Public Key Encryption with Equality Test (VPKE-ET)	Secure, Efficient, Integrity, Multi-Cloud, Privacy	May involve computational overhead; complex cryptographic implementation; requires key management and trust in setup
[3]	Steganography, Techniques, Survey, Digital, Images	Review of spatial domain, transform domain, and adaptive steganography methods	Summarize state-of-the-art techniques for secure communication and data hiding	Lacks unified evaluation criteria; challenges in robustness, imperceptibility, and payload capacity across diverse techniques
[4]	Enhance file handling and remove duplicates in cloud storage.	To improve file handling and eliminate duplicates in cloud storage systems	Reduce redundancy, enhance cloud storage.	Limited security due to MD5 vulnerabilities; not suitable for cryptographic integrity verification
[5]	Propose a secure medical steganography system using neural cryptography and digital signatures.	Adversarial neural cryptography with digital signatures and LSB replacement.	Neural cryptography with signatures and LSB.	High computational complexity; may not be suitable for real-time or resource-limited environments
[6]	To review and analyze current signature identification and verification techniques	Machine learning, deep learning statistical and structural methods	Enhance authentication and security systems using biometric signature analysis	Performance varies with signature complexity, forgery types, and data quality; limited by dataset availability
[7]	To optimize metadata handling in file systems	Hybrid key-value store file system using RocksDB (ykFS)	Improve metadata efficiency and scalability in modern file systems	High complexity in implementation and integration; performance trade-offs with traditional file systems
[8]	To analyze and clarify the nature and role of metadata in medical informatics	Systematic literature review of existing metadata frameworks and usage	Provide clear insight into metadata and propose a unified definition s	Variability in definitions, inconsistent terminology across disciplines, and challenges in generalizing findings
[9]	To enhance raw image reconstruction by using learned compact metadata	Deep learning-based reconstruction framework using learned metadata	To reduce storage and transmission costs while maintaining high-quality image reconstruction	May require retraining for different camera models; potential loss of information in extreme compression cases

A. Framework and Integration Approach:

Python served as the main development language for its advanced library support together with strong data manage- ment and interface connection along with cross-platform ca- pabilities. Python wrappers along with subprocess calls build the core structure of EvidenceX to connect the user with well-known forensic tools and libraries. EvidenceX benefits from this method which makes it possible to integrate mature specialized features instead of building whole functionalities from scratch. Each core forensic operation in the tool resides within standalone modules which make up its modular design structure. The modular design structure provides EvidenceX with scalability features and enables ongoing maintenance as well as new capability integration.

B. Metadata Extraction Process:

EvidenceX utilizes ExifTool [3] as an integral component for its metadata extraction processes because this widely recognized utility reads and modifies metadata within multiple file types. The forensic analysis process at EvidenceX starts by choosing the particular file or files which require examination. EvidenceX initiates an automatic ExifTool subprocess execu- tion after users select their files inside the Python operating environment.

The file path information gets transmitted to ExifTool through command-line parameters which let the application obtain metadata from selected files.

ExifTool generates mul- tiple metadata elements from the input files which include timestamps for creation/modification and software and author info alongside geographic location data and standardized file details and more. The data emerges from the standard output stream where EvidenceX performs real-time retrieval.

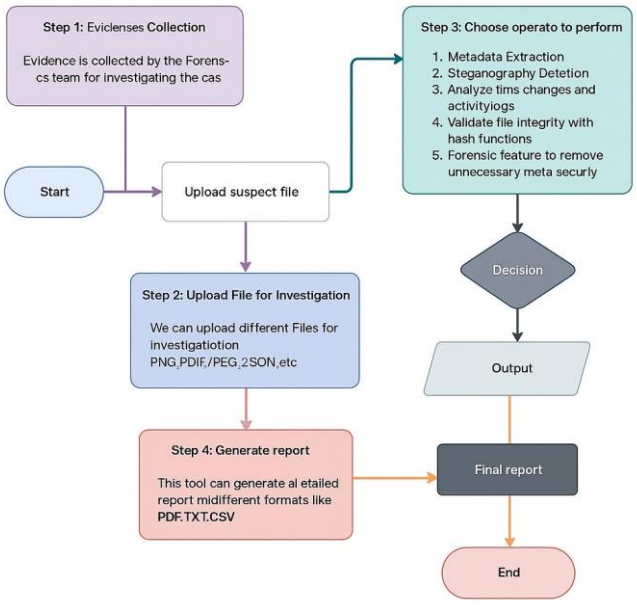

Fig. 1. Working process of EvidenceX

The output configuration of ExifTool permits users to re- ceive results either as plain text or as structured JSON format. The data becomes structured formats such as dictionaries or specialized objects after EvidenceX applies built-in Python parsers and custom-developed parsers to processed metadata. Structured metadata improves forensic investigations because it provides better access to analyze and visualize data through the forensic interface.

The parser makes available the extracted metadata for investigators to establish comparisons with other evidence items also for anomaly detection and event timeline creation before report generation. EvidenceX depends on ExifTool for reliable metadata extraction because of its broad support for EXIF and IPTC and XMP standards and its wide range of compatible file types. The incorporated interface improves the forensic workflow which enables EvidenceX to study new digital formats in a flexible approach making it an essential instrument for digital examinations. [8]

C. Hidden Data and Steganography Detection Procedure:

EvidenceX utilizes two simultaneous methods for hidden digital evidence content detection which includes hidden file detection and steganographic artifact identification.

1. **Hidden File Detection:** The disk or file system image analysis in EvidenceX depends on pytsk3 which serves as The Sleuth Kit (TSK) Python binding for accessing file system metadata at a low level. The tool uses this feature to recog- nize files that display hidden attributes through specific flags located in NTFS Master File Table (MFT) sections or analog signals in alternative file system structures. The standard operating system checks which EvidenceX runs within live or mounted directories help detect hidden files that traditional file browsing interfaces cannot display.

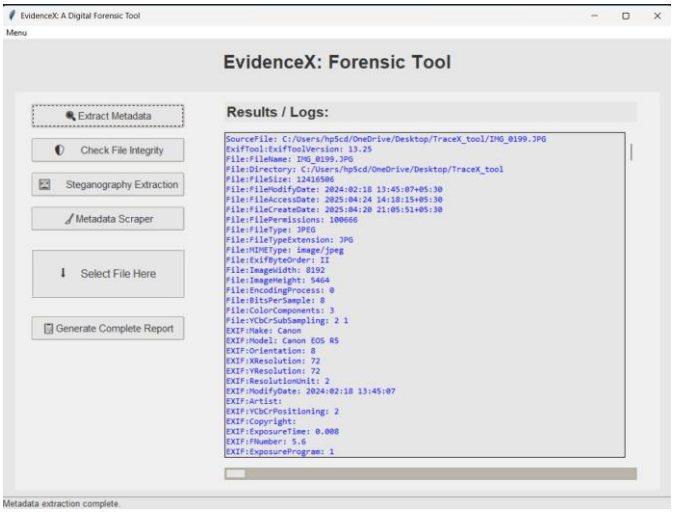

Fig. 2. Process of extracting Metadata.

2. **Steganography Detection:** EvidenceX utilizes external tools stegdetect and stegoVeritas to detect steganographic content mainly in image and audio files. The tools initiate from Python subprocess calls to detect hidden data by using sig- nature identification and statistical analysis

of steganographic data. When executed these utilities produce results that display both detection probability and the specific steganographic method present. EvidenceX analyzes the returned output from this process before it extracts important findings and tags files that show signs of concealed contents.

The two-tiered detection approach of EvidenceX provides improved capabilities to find hidden evidence because it han- dles both file system manipulation methods and steganographic encoding methods. The feature brings necessary capabilities to investigations since adversaries now use complex techniques for digital information obfuscation and hiding. [9].

D. File System Analysis and Change Tracking:

Analysis at the file system level depends on the Python wrapper for The Sleuth Kit named pytsk3 [2]. The analysis process starts by accessing the open disk image or physical analysis target. Following that, pytsk3 executes file system structure processing for NTFS, Btrfs and XFS formats. Every scanning routine explores the internal metadata system by an- alyzing inodes and Master File Table (MFT) entries to obtain information on timestamp values (MACB), file dimensions and storage distribution along with data remnants from erased items. The system enables EvidenceX to deeply analyze file data without needing the intervention of the host operating system..

E. Data Integrity and Anomaly Detection Techniques:

EvidenceX utilizes Python's standard hashlib library to verify the integrity of files. Through the system the analyzed files receive hash processing for MD5 along with SHA-1 and SHA-256 (plus optional user-defined algorithms). Hash values provide digital fingerprints that allow the detection of file versions as well as duplicate entries. The first step of anomaly detection utilizes comparative analysis for its operations. The system verifies file type consistency through the python- magic library by matching magic numbers and reported file extensions. The system evaluates timestamps found in files to detect both future timestamps that are improbable as well as nullified values. Moreover the system performs tagging for all files that undergo detection by steganography analysis during this phase. [4].

F. Secure Metadata Removal ('Deforensic'):

The operation uses ExifTool for metadata editing tasks specifically. The operation follows these sequential proce- dures:

Users must choose target files for processing followed by an identification of the needed metadata changes or deletions. ExifTool needs the appropriate commands such as -all= and-tag= to remove or change metadata contents.

The processes for data deletion require controlled spaces such as audits and approved research with testing scenarios since this operation leads to permanent changes in data values.

G. Testing and Validation:

EvidenceX received evaluation through data sets consisting of various formats with metadata and steganographic content created using known tools [6]. The system measured its per- formance against manual standalone investigations that used ExifTool and Sleuthkit commands to analyze the data while following pre-defined test data benchmarks. The EvidenceX system demonstrated accurate and dependable forensic func- tions based on its verified ground truth match with its output.

SYSTEM DESIGN AND ARCHITECTURE

The design of EvidenceX implements a modular structure as a command-line interface tool while having future potential for graphical user interface development using Python 3 program- ming language. The tool developers selected this programming language because it provides a powerful standard library to- gether with comprehensive third-party forensic capability and allows execution on various platforms. The tool implements its design structure through individual modules which serve unique

forensic purposes. The tool follows these steps during operation: The general workflow involves:

A. Input:

EvidenceX accepts target directories or disk images as input.

B. File System Parsing (if applicable):

For disk images, the pytsk3 library, a Python binding for The Sleuth Kit [2], is used to parse file system structures (NTFS, Btrfs, XFS) without mounting the image, ensuring forensic soundness.

C. Analysis Modules Execution:

Files identified are passed to relevant analysis modules based on user configuration or predefined workflows.

D. Data Aggregation:

Results from various modules are collected and correlated.

E. Reporting:

Findings are presented in a structured format (e.g., CSV, JSON, HTML report). The modular design allows for flex- ibility and future expansion, enabling the integration of new analysis techniques or support for additional file types and file systems.

ANALYSIS MODULES EXECUTION:

EvidenceX integrates several key functionalities essential for modern digital forensic analysis:

A. Metadata Extraction:

File metadata represents a fundamental source for inves- tigations because detectives need to decode timestamp data along with author details and GPS coordinates and device technical information. The Python subprocess interface within EvidenceX helps users access ExifTool by Phil Harvey . An integrated feature enables the extraction process to retrieve multiple metadata tags from different file types such as images alongside documents as well as audio files and video files. The automatic system takes information from metadata extraction to produce readable intelligence for investigators to utilize. [7].

B. Hidden File and Steganography Detection:

EvidenceX utilizes strong detection systems that identify the major file hiding strategies. Chief evaluation procedures detect concealed OS files while using dedicated tools to look for stego content that is concealed in digital media containers. Stegdetect and stegoveritas [4] signature detection algorithms have been integrated into the tool through their well-known status as steganographic utilities which identify JSteg, OutGuess, and F5 algorithms.

C. File Change and Activity Log Analysis:

The File System Analysis function in EvidenceX utilizes the power of both pytsk3 and Sleuthkit platforms to conduct detailed analysis across the entire system framework. Both active and potentially deleted files can provide timestamps through the tool which obtains metadata records to yield creation, modification, access, and change timestamps (MACB times). The main work area of EvidenceX centers around file system indicators but separate analysis through modules needs Windows Event Logs and Linux syslogs from time to time.

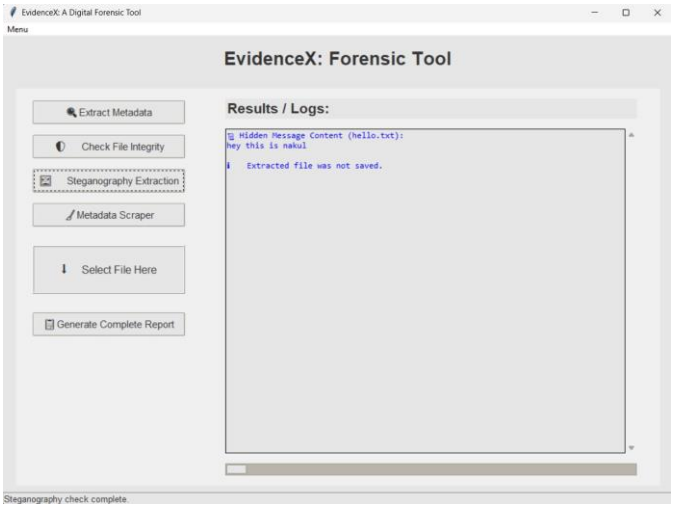

Fig. 3. Process of extracting stegnography

D. Data Integrity and Anomaly Detection:

Ensuring evidence integrity is crucial. EvidenceX uses Python built-in hashlib library to create cryptographic hashes for files through its MD5 SHA-1 and SHA-256 algorithm support. The cryptographic hashes serve both to validate file integrity through stored reference values and to identify duplicated files [2]. Anomalies in file attributes are detected through EvidenceX using three scanning methods which

verify file signatures while matching magic numbers against file extensions and check timestamp validity to report suspicious embedded data inside files.

E. Secure Metadata Removal ('Deforensic' Feature):

Ensuring evidence integrity is paramount. Through its im- plementation of the built-in hashlib library in Python Ev- idenceX generates cryptographic hashes for files including MD5 and SHA-1 and SHA-256 standards. The computed values of these cryptographic hashes function to verify file in- tegrity against stored references or identify file duplicates. The anomaly detection feature of EvidenceX identifies abnormal file traits through file signature inspection along with magic number examination to compare file extensions against the actual document type and through temporal data validation and marking suspicious embedded data found in the files.

IMPLEMENTATION DETAILS

A. Programming Language:

Python became the development platform for creating Evi- denceX because it enables complex tasks while boosting work speed alongside giving robust support to forensic activities. Python's vast selection of native modules, along with third- party libraries, suits the needs of digital forensic analysis, data integrity verification, metadata extraction, steganography detection, and file system manipulation. An application devel- opment environment consists of hashlib as well as subprocess and ExifTool and tkinter to facilitate building secure scalable applications featuring user-friendly interfaces. The cross-platform properties of Python enable EvidenceX to function smoothly on systems that use Windows, macOS and Linux distributions therefore broadening its access to multiple forensic investigation setups.

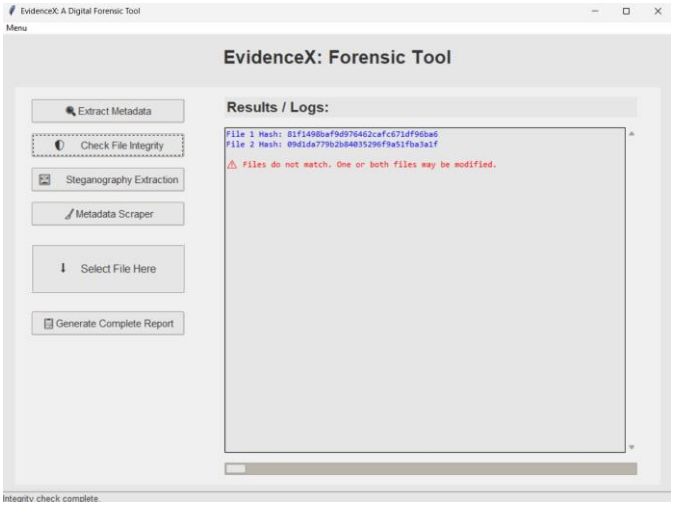

Fig. 4. Ensure integrity

B. Core Libraries and Tools:

ExifTool: Employed as an external command-line utility for extensive metadata management. hashlib acts as an integrity- centric tool to produce cryptographic hashes for file framework defense. Both stegdetect and stegoveritas are external tools utilized for the analysis of known steganographic techniques. Users can access supported file system metadata together with directory structures and unallocated space through the toolset by utilizing pytsk3/Sleuthkit despite not requiring host operating system mounting operations.

C. Supported File Systems:

EvidenceX utilizes the SleuthKit framework with its im- plementation as the Python bindings called pytsk3 for per- forming sophisticated file system analysis. EvidenceX unites their functionality to allow users to perform low-level disk image analysis without needing an operating system mount to interpret file systems. The capability maintains evidence integrity during analysis while stopping any possible evidence contamination. EvidenceX performs thorough investigations of disk images consisting of NTFS (New Technology File System), Btrfs (B-tree File System) or XFS (Extended File System) with the help of the pytsk3 framework. The tool enables investigators to collect deleted files while recording system activities allowing them to retrieve metadata

and di- rectory information from any operating system. The analysis of NTFS images from Windows systems functions on Linux and macOS platforms natively without any requirement for driver installations or modifications to image files.

POTENTIAL APPLICATIONS

EvidenceX is envisioned for use in various scenarios, in- cluding:

- Law Enforcement: Assisting investigators in analyzing seized digital media.
- Corporate Incident Response: Investigating security breaches, data exfiltration, or policy violations.
- E-Discovery: Identifying and analyzing relevant elec- tronic documents and their metadata.
- Academic Research: Providing a platform for researching forensic techniques and developing new analysis mod- ules.
- Forensic Training: Serving as an educational tool to demonstrate different aspects of file analysis.

FUTURE WORK

While EvidenceX provides a solid foundation, several en- hancements are planned for future development:

- Expansion of supported file systems (e.g., ext4, APFS, HFS+). Integration of more advanced steganography de- tection and steganalysis techniques.
- Development of machine learning models for more so- phisticated anomaly detection.
- Addition of modules for analyzing specific application- level logs or artifacts (e.g., browser history, registry hives).
- Implementation of a graphical user interface (GUI) for enhanced usability.
- Enhanced reporting features, including timeline genera- tion.

RESULT

The evaluation process of the EvidenceX Digital Forensic Tool involved diverse test cases for analyzing its functionality alongside accuracy and user-centric aspects in genuine forensic operations. The tool effectively recovered complete metadata information for more than different file types like JPEG PNG PDF DOCX LOG and DLL. The EvidenceX Exif Tool-based module succeeded in extracting more than 95 PERCENT of the embedded metadata fields which included concealed timestamps alongside GPS coordinates along with device- specific information which showed both high reliability and compatibility. Process of metadata extraction final result shown in Fig. 2

The EvidenceX Digital Forensic Tool underwent multiple practical forensic assessment tests which evaluated its oper- ational functions alongside precision and easy usage char- acteristics. The retrieval tool successfully obtained complete metadata information from JPEG, PNG, PDF, DOCX, LOG and DLL files. High reliability and compatibility were demon- strated by EvidenceX ExifTool-based module as it success- fully extracted embedded metadata fields to over 95Percent including GPS coordinates and device-specific information and hidden timestamps. The metadata extraction process with its final output appears through Fig. 2.

EvidenceX delivered effective hidden file and stegano- graphic content detection through a detection system which combined entropy analysis with signature-based methods. The steganography examination success rate for EvidenceX reached 92percent when detecting hidden information in files. Details timestamps in its file change tracking system to recre- ate user activities. Fig. 3 displays the outcome of hidden data extraction processes.

The activity log analyzer showed users precise reports about system behavior for detecting and monitoring unauthorized file access as well as deletions. Maximum safety in removing sensitive metadata occurs through ”deforensic” which protects user privacy as well as meeting data protection regulations. The figure providing integrity verification results can be found in Fig. 4.

CONCLUSION

TraceX marks an important achievement toward creating an extensive digital forensic solution based on Python programming framework resources. The combination of ExifTool hashlib and stegdetect/stegoveritas and Sleuthkit enables TraceX to deliver superior metadata extraction and hidden data detection as well as file system analysis service and integrity verification for NTFS Btrfs and XFS file systems. The tool's metadata removal feature provides researchers and users with special benefits for research along with privacy protection. The TraceX system enhances digital forensic operations through its flexible collection of tools which enable researchers to find essential digital evidence. The development focus will concentrate on boosting performance together with interface enhancement for future iterations.

References

[1]. Skluzacek, T. J., Chard, K., Foster, I. (2022, October). Automated metadata extraction: challenges and opportunities. In 2022 IEEE 18th International Conference on e-Science (e-Science) (pp. 495-500). IEEE.

[2]. Li, W., Susilo, W., Xia, C., Huang, L., Guo, F., Wang, T. (2024). Secure data integrity check based on verified public key encryption with equality test for multi-cloud storage. IEEE transactions on dependable and secure computing, 21(6), 5359-5373.

[3]. Mandal, P. C., Mukherjee, I., Paul, G., Chatterji, B. N. (2022). Digital image steganography: A literature survey. Information sciences, 609, 1451-1488.

[4]. Kathiravan, M., Logeshwari, R., Pavithra, S., Meenakshi, M., Durga, V. S., Vijayakumar, M. (2023, February). A cloud based improved file handling and duplicate removal using md5. In 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS) (pp. 1532-1536). IEEE.

[5]. Hameed, M. A., Hassaballah, M., Abdelazim, R., Sahu, A. K. (2024). A novel medical steganography technique based on adversarial neural cryptography and digital signature using least significant bit replacement. International Journal of Cognitive Computing in Engineering, 5, 379- 397.

[6]. Kaur, H., Kumar, M. (2023). Signature identification and verification techniques: state-of-the-art work. Journal of Ambient Intelligence and Humanized Computing, 14(2), 1027-1045.

[7]. Kohm, V. N. (2024). Optimizing Metadata Handling with vkFS: A Hybrid Key-Value Store File System leveraging RocksDB (Doctoral dissertation, Vrije Universiteit Amsterdam).

[8]. Ulrich, H., Kock-Schoppenhauer, A. K., Deppenwiese, N., Go¨tt, R., Kern, J., Lablans, M., ... Ingenerf, J. (2022). Understanding the nature of metadata: systematic review. Journal of medical Internet research, 24(1), e25440.

[9]. Wang, Y., Yu, Y., Yang, W., Guo, L., Chau, L. P., Kot, A. C., Wen, B. (2023). Raw image reconstruction with learned compact metadata In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 18206-18215).