

DESIGN AND EVALUATION OF
AN AUTOMATED FRAMEWORK
FOR DETECTING ANTI-
FORENSIC MANIPULATION IN
MODERN FILE SYSTEMS

RESEARCH CONTEXT/BACKGROUND

Broader context

Digital forensic investigations increasingly face intentional evidence manipulation, not just concealment.

In a traditional sense forensic analysis assumed:

- Artifacts were incomplete or partially detected
- Evidence degradation was passive

However modernized attackers deploy anti-forensic techniques which are designed to:

- Undermine evidential integrity
- Introduce reasonable doubt in court
- Delay incident response
- Obstruct attribution

Emerging Threat: Anti forensics

- Timestamp manipulation (Timestomping)
- Secure deletion/ wiping tools
- Increasing sophistication in evidence concealment



RESEARCH PROBLEM

The Core problem

Current forensic workflows follow these three points:

- Extract artifacts efficiently
- Do not automatically evaluate artifact integrity
- Do not systematically score manipulation likelihood

This then leads to three major challenges:

1. Investigator Dependence
2. Fragmented Artifact analysis
3. Lack of Quantified Tampering Assessment

Problem Statement

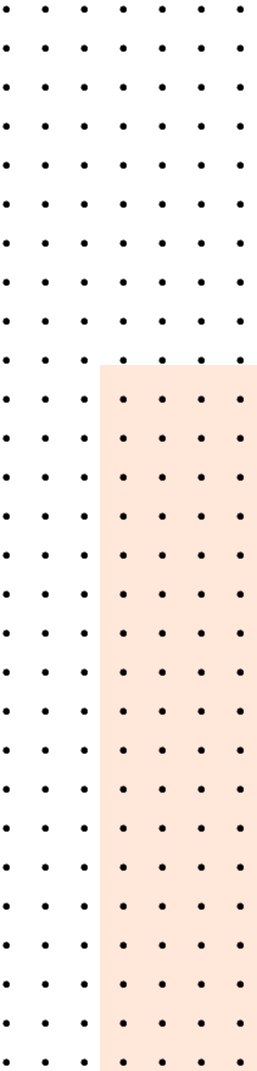
There is no lightweight automated NTFS-focused framework that detects and quantifies timestamp manipulation and secure deletion artefacts together.



LITERATURE REVIEW

Anti-forensic techniques such as secure deletion and timestamp manipulations are a significant challenge to NTFS-based digital forensic investigations.

Research clearly shows that secure wiping tools often leave behind detectable artefacts, which include abnormal cluster allocation patterns and entropy inconsistencies.[1]Timestamp manipulation or (Time stamping) can be identified through inconsistencies between NTFS metadata attributes, though reliability varies depending on the system conditions[2]. Enhanced detection approaches using volume shadow copies and structured metadata analysis improve identification accuracy[3,4]. However, the existing research largely examines these techniques independently and often requires manual intervention and interpretation. A clear gap remain in developing an automated, integrated NTFS forensic framework capable of correlating multiple anti-forensic behaviours and systematically assessing the likelihood of tampering.[5]

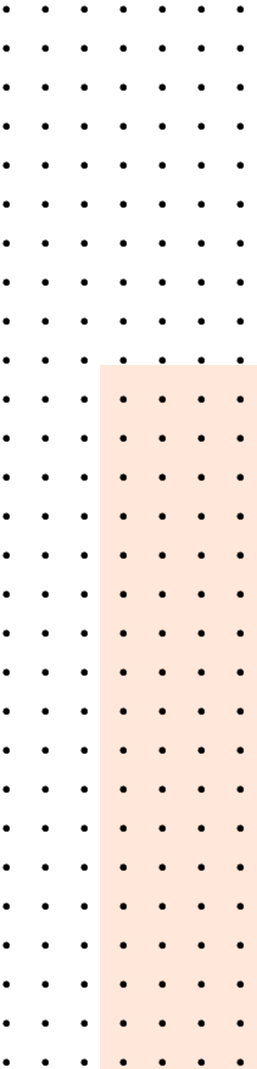


RESEARCH GAP

The literature demonstrates that secure deletion leaves detectable artefacts [1], timestamp manipulation can be identified via metadata inconsistencies [2]–[4], and anti-forensic techniques are increasingly prevalent and sophisticated [5]. However, several limitations remain:

1. Detection techniques are typically studied in isolation (wiping versus timestamp manipulation).
2. Automation is limited or tool-specific.
3. Few studies propose structured scoring models to quantify tampering likelihood.
4. Integrated evaluation across multiple anti-forensic scenarios is underexplored.

Accordingly, while individual detection mechanisms exist, there remains a gap in developing a unified, automated NTFS-focused framework that combines timestamp anomaly detection and secure deletion artefact analysis within a structured evaluation model.



RESEARCH AIMS AND OBJECTIVES

Aim:

To design and evaluate an automated framework for detecting timestamp manipulation and secure deletion artefacts in NTFS file systems

Objectives:

1. Analyse NTFS timestamp structures
2. Simulate time stamping and secure deletion
3. Develop automated detection models
4. Implement weighted anomaly scoring
5. Evaluate detection accuracy and reliability

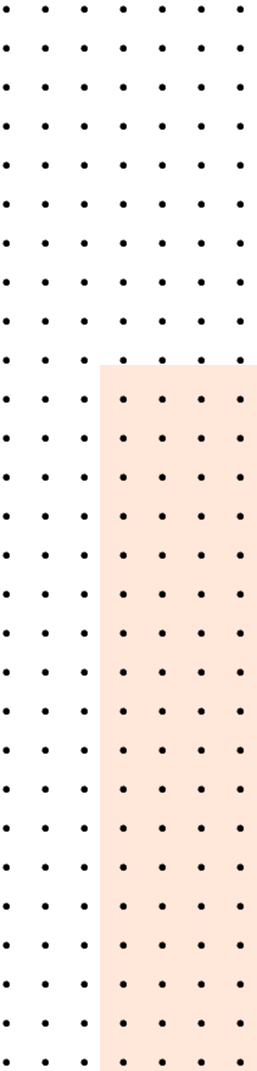


M E T H O D O L O G Y

Research Design Approach

Structure:

1. Problem identification
2. Artefact design
3. Implementation
4. Evaluation
5. Analysis



Data Collection

- Establish baseline (normal file activity)
- Execute controlled anti-forensic actions:
 - Timestamp manipulation
 - Secure deletion
- Extract NTFS metadata (\$STANDARD_INFORMATION, \$FILE_NAME)
- Analyse timestamp inconsistencies, entropy levels, and cluster allocation patterns

Framework Development

- Implement automated metadata comparison
- Detect wipe artefacts
- Apply weighted anomaly scoring model
- Generate structured tampering assessment output

Evaluation

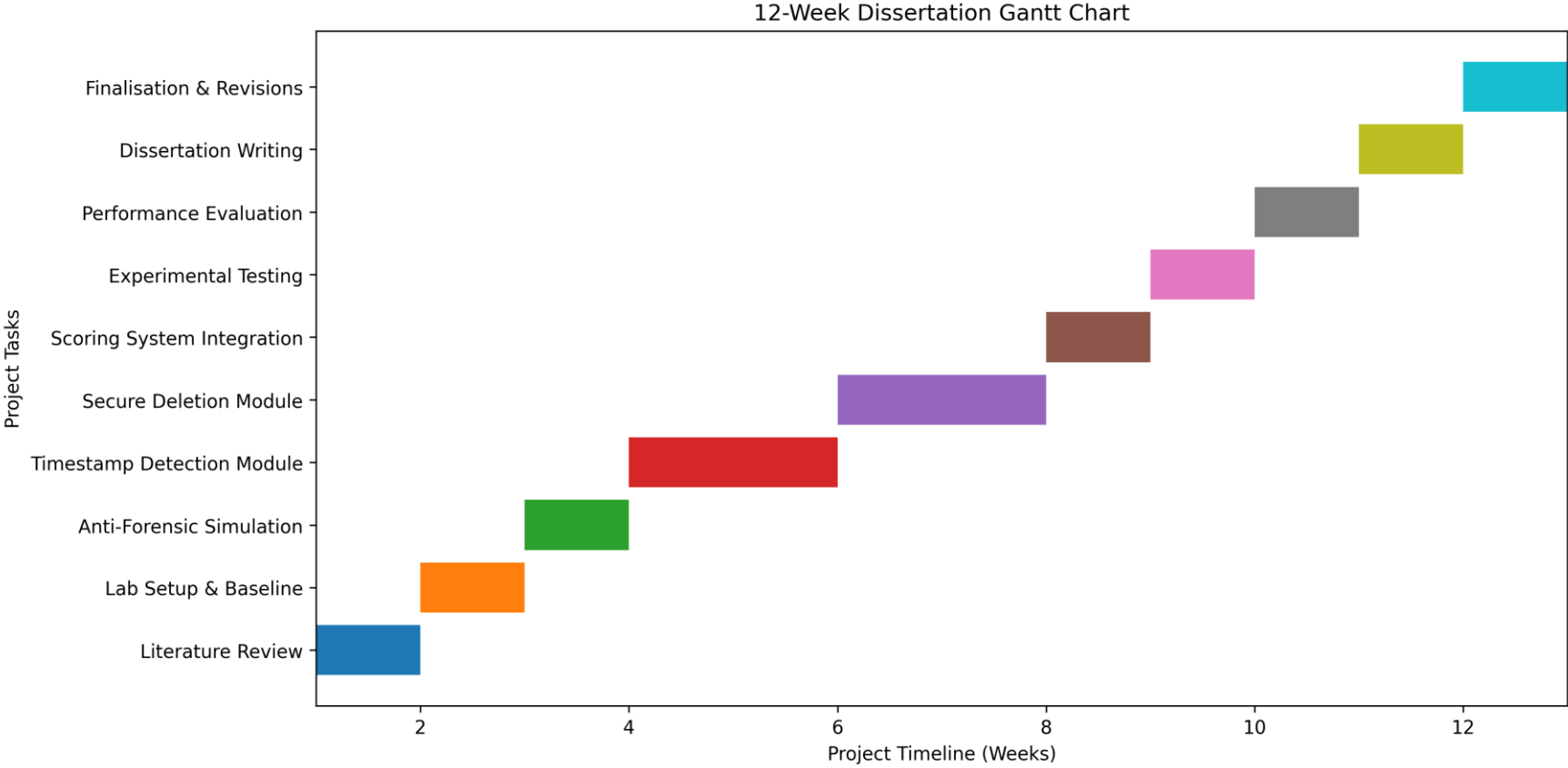
- Test across normal and tampered scenarios
- Measure accuracy, precision, recall, and false positives



SUMMARY TABLE

Category	Component	Purpose in Project	
Hardware	High-spec PC (16GB+ RAM)	Run VM & process forensic images	
Storage	External drive	Preserve disk images & ensure reproducibility	
Virtualisation	VirtualBox / VMware Workstation	Controlled NTFS test environment	
Operating System	Windows 10/11 (NTFS)	Realistic file system environment	
Anti-Forensic Simulation	Timestomp	Manipulate NTFS timestamps	
	SDelete	Simulate secure overwrite behaviour	
	FTK Imager	Create forensic disk images	• • •
Forensic Imaging	Autopsy	Ground truth verification	• • •
Validation Tool	Python + pytsk3	NTFS metadata extraction	• • •
Development			• • •
Data Analysis	numpy / pandas	Process & analyse anomaly data	• • •
Visualisation	matplotlib	Present evaluation metrics	• • •
Methodology	Design Science Research	Artefact development & evaluation	• • •
			• •
Experimental Design	Controlled baseline vs manipulated images	Enable measurable comparison	• •
Detection Model	Rule-based + weighted scoring	Quantify tampering likelihood	• •
Evaluation Metrics	Accuracy, Precision, Recall, FPR	Measure framework performance	• •
			• •
			• •
			• •
			• •

PROJECT PLANNING AND RESOURCES



PROJECT IMPACTS



Advancement of Digital Forensic Automation

Contributes toward automated detection of anti-forensic behaviour in NTFS environments.



Improved Investigator Efficiency

Reduces reliance on manual metadata comparison and artefact correlation.



Structured Tampering Assessment Model

Introduces a scoring-based approach to quantify likelihood of manipulation.



Enhanced Evidence Reliability

Supports stronger forensic reporting by identifying inconsistencies systematically.



Academic Contribution

Bridges the gap between isolated timestamp and wipe detection research.



Foundation for Future Research

Provides a scalable model that could be extended to other file systems or additional anti-forensic techniques.

PROJECT LIMITATIONS



NTFS-Specific Scope

The framework is limited to NTFS file systems and may not generalize to FAT32, exFAT, or ext4 environments.



Dependence on Artefact Availability

Detection accuracy relies on the presence of metadata inconsistencies or wipe artefacts. If artefacts are thoroughly removed, detection may be reduced.



Limited Anti-Forensic Coverage

The project focuses only on timestamp manipulation and secure deletion, excluding other anti-forensic techniques (e.g., encryption, steganography, log tampering).



Controlled Experimental Environment

Testing is conducted in a virtualized lab environment, which may not fully replicate complex real-world enterprise systems.



Time Constraints (3-Month Scope)

The project prioritizes a lightweight, proof-of-concept implementation rather than a production-ready forensic tool.



Potential False Positives

Legitimate system processes (e.g., file copying, system updates) may create metadata anomalies similar to tampering behavior.

REFERENCES

- [1] Y. Li and X. Cui, "De-Wipimization: Detection of data wiping traces for investigating NTFS file system," *Computers & Security*, vol. 99, 2020. <https://doi.org/10.1016/j.cose.2020.102034>
- [2] R. Buchholz and E. H. Spafford, "Artifacts for Detecting Timestamp Manipulation in NTFS on Windows and Their Reliability," *Forensic Sci. Int.: Digital Investigation*, vol. 32, 2020. <https://doi.org/10.1016/j.fsidi.2020.300920>
- [3] M. ALJI and K. CHOUGDALI, "Detection of Suspicious Timestamps in NTFS using Volume Shadow Copies," *Int. J. Comp. Net. Info. Sec.*, vol. 13, no. 4, 2021. <https://doi.org/10.5815/ijcnis.2021.04.06>
- [4] J. Oh, S. Lee and H. Hwang, "Forensic Detection of Timestamp Manipulation for Digital Forensic Investigation," *IEEE Access*, vol. 12, 2024. <https://doi.org/10.1109/ACCESS.2024.3395644>
- [5] R. González Arias et al., "Systematic Review: Anti-Forensic Computer Techniques," *Appl. Sci.*, vol. 14, no. 12, 2024. <https://doi.org/10.3390/app14125302>

