

# Algorithmic Techniques for Information Hiding and Digital Object Protection

A Dissertation

submitted to the designated  
by the Assembly  
of the Department of Computer Science and Engineering  
Examination Committee

by

Vasileios Vouronikos

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

University of Ioannina

School of Engineering

Ioannina 2026

Advisory Committee:

- **Stavros D. Nikolopoulos**, Full Professor, Department of Computer Science and Engineering, University of Ioannina (Supervisor).
- **Loukas Georgiadis**, Full Professor, Department of Computer Science and Engineering, University of Ioannina
- **Emmanouil Magkos**, Full Professor, Department of Informatics, Ionian University

Examining Committee:

- **Stavros D. Nikolopoulos**, Full Professor, Department of Computer Science and Engineering, University of Ioannina (Supervisor).
- **Loukas Georgiadis**, Full Professor, Department of Computer Science and Engineering, University of Ioannina.
- **Emmanouil Magkos**, Full Professor, Department of Informatics, Ionian University.
- **Euripides Markou**, Full Professor, Department of Computer Science and Engineering, University of Ioannina.
- **Michael A. Bekos**, Associate Professor, Department of Mathematics, University of Ioannina.
- **Yiannis Stamatiou**, Full Professor, Business Administration Department, University of Patras.
- **Christos Tjortjis**, Full Professor, Department of Science and Technology, International Hellenic University.

# ACKNOWLEDGEMENTS

---

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Stavros Nikolopoulos, for his invaluable guidance, patience, and trust throughout the course of my doctoral research. His advice and expertise were instrumental in the completion of this work. I would also like to thank the members of my advisory committee, Professor Emmanouil Magkos, Professor Leonidas Palios and Professor Loukas Georgiadis, for their insightful feedback and guidance throughout my research. I am grateful to the members of the examination committee, Professor Euripides Markou, Associate Professor Michael Bekos, Professor Yannis Stamatiou and Professor Christos Tjortjis for the time they dedicated to reviewing my dissertation and for their constructive comments during my presentation. A heartfelt thank you goes to my parents Thodoris and Stauroula, for their unconditional support, patience, and love throughout this journey. None of this would have been possible without them. I would also like to thank my siblings Nikos and Vivi, who were always by my side with encouragement and understanding. Finally, I would like to thank my beloved friends Sifis, Zoitsa, Kostas, Maria, Eirini, Stelios, Alexis, Filio, for their support, encouragement, and the moments of relaxation they offered whenever I needed them most.

# TABLE OF CONTENTS

---

|                                                                             |           |
|-----------------------------------------------------------------------------|-----------|
| List of Figures                                                             | v         |
| List of Tables                                                              | ix        |
| List of Algorithms                                                          | xiv       |
| Abstract                                                                    | xv        |
| Εκτεταμένη Περίληψη                                                         | xviii     |
| <b>1 Introduction</b>                                                       | <b>1</b>  |
| 1.1 Digital Object Types . . . . .                                          | 1         |
| 1.2 Digital Object Security . . . . .                                       | 2         |
| 1.3 Information Hiding . . . . .                                            | 4         |
| 1.4 Authenticity and Integrity Preservation . . . . .                       | 7         |
| 1.5 Thesis Contributions . . . . .                                          | 8         |
| 1.6 Structure of the Thesis . . . . .                                       | 10        |
| <b>2 Theoretical Background</b>                                             | <b>12</b> |
| 2.1 Related Work on Information Hiding and Watermarking . . . . .           | 12        |
| 2.2 Information Hiding in the Frequency Domain . . . . .                    | 13        |
| 2.3 Information Hiding in the Spatial Domain . . . . .                      | 15        |
| 2.4 Related Approaches in Authenticity and Integrity Preservation . . . . . | 16        |
| 2.4.1 Frequency Based Watermarking Techniques . . . . .                     | 18        |
| 2.4.2 Spatial Based Watermarking Techniques . . . . .                       | 19        |
| 2.4.3 ML-based Watermarking Techniques . . . . .                            | 20        |
| 2.4.4 Hybrid Watermarking Techniques . . . . .                              | 22        |



|          |                                                            |           |
|----------|------------------------------------------------------------|-----------|
| <b>3</b> | <b>Components and Functional Characteristics</b>           | <b>24</b> |
| 3.1      | Digital Objects Characteristics . . . . .                  | 24        |
| 3.1.1    | Image Characteristics . . . . .                            | 25        |
| 3.1.2    | Audio Characteristics . . . . .                            | 25        |
| 3.1.3    | Video Characteristics . . . . .                            | 26        |
| 3.1.4    | PDF Characteristics . . . . .                              | 26        |
| 3.2      | Digital Objects Segmentation Rules . . . . .               | 27        |
| 3.2.1    | Spatial Segmentation . . . . .                             | 27        |
| 3.2.2    | Temporal Segmentation . . . . .                            | 31        |
| 3.3      | Frequency Domain Watermarking . . . . .                    | 33        |
| 3.4      | Spatial Domain Watermarking . . . . .                      | 34        |
| 3.5      | Encoding and Hashing Techniques . . . . .                  | 36        |
| 3.5.1    | Encoding Integers to Self-Inverting Permutations . . . . . | 36        |
| 3.5.2    | Integrity Preservation Utilizing Hash Function . . . . .   | 38        |
| <b>4</b> | <b>Models: Authenticity and Integrity</b>                  | <b>40</b> |
| 4.1      | Authenticity: Embedding Robust Information . . . . .       | 40        |
| 4.1.1    | Images . . . . .                                           | 40        |
| 4.1.2    | Audio . . . . .                                            | 57        |
| 4.1.3    | Video . . . . .                                            | 60        |
| 4.1.4    | Documents . . . . .                                        | 67        |
| 4.2      | Integrity: Embedding Fragile Information . . . . .         | 75        |
| 4.2.1    | Images . . . . .                                           | 75        |
| 4.2.2    | Audio . . . . .                                            | 80        |
| 4.2.3    | Videos . . . . .                                           | 84        |
| 4.2.4    | Documents . . . . .                                        | 88        |
| <b>5</b> | <b>Architecture: Integration and Security</b>              | <b>90</b> |
| 5.1      | Integration Procedure . . . . .                            | 90        |
| 5.2      | Image Authenticity and Integrity . . . . .                 | 90        |
| 5.3      | Audio Authenticity and Integrity . . . . .                 | 92        |
| 5.4      | Video Authenticity and Integrity . . . . .                 | 94        |
| 5.5      | Document Authenticity and Integrity . . . . .              | 95        |

|          |                                                      |           |
|----------|------------------------------------------------------|-----------|
| <b>6</b> | <b>Evaluation</b>                                    | <b>97</b> |
| 6.1      | Data-set Description . . . . .                       | 97        |
| 6.2      | Quality, Robustness and Fragility Metrics . . . . .  | 101       |
| 6.2.1    | Quality and Imperceptibility Measurement . . . . .   | 102       |
| 6.2.2    | Robustness Measurement . . . . .                     | 104       |
| 6.2.3    | Fragility Measurement . . . . .                      | 105       |
| 6.3      | Computational Efficiency Metrics . . . . .           | 107       |
| 6.4      | Attack Vector . . . . .                              | 108       |
| 6.4.1    | Filtering Attacks . . . . .                          | 108       |
| 6.4.2    | Compression Attacks . . . . .                        | 109       |
| 6.4.3    | Geometric and Document-processing Attacks . . . . .  | 109       |
| 6.4.4    | Content Modification and Tampering Attacks . . . . . | 110       |
| 6.4.5    | Temporal Attacks . . . . .                           | 110       |
| 6.4.6    | AI-based Attacks . . . . .                           | 111       |
| 6.4.7    | Combined Attacks . . . . .                           | 111       |
| 6.5      | Experimental Results on Images . . . . .             | 112       |
| 6.5.1    | Images: Run-time Efficiency . . . . .                | 116       |
| 6.5.2    | Images: Quality and Imperceptibility . . . . .       | 121       |
| 6.5.3    | Images: Authenticity Verification . . . . .          | 125       |
| 6.5.4    | Images: Integrity Attestation . . . . .              | 134       |
| 6.5.5    | Images: Observations . . . . .                       | 136       |
| 6.6      | Experimental Results on Audio . . . . .              | 138       |
| 6.6.1    | Audio: Run-time Efficiency . . . . .                 | 140       |
| 6.6.2    | Audio: Quality and Imperceptibility . . . . .        | 142       |
| 6.6.3    | Audio: Authenticity Verification . . . . .           | 143       |
| 6.6.4    | Audio: Integrity Attestation . . . . .               | 144       |
| 6.6.5    | Audio: Observations . . . . .                        | 145       |
| 6.7      | Experimental Results on Video . . . . .              | 146       |
| 6.7.1    | Video: Run-time Efficiency . . . . .                 | 149       |
| 6.7.2    | Video: Quality and Imperceptibility . . . . .        | 153       |
| 6.7.3    | Video: Authenticity Verification . . . . .           | 156       |
| 6.7.4    | Video: Integrity Attestation . . . . .               | 164       |
| 6.7.5    | Video: Observations . . . . .                        | 166       |
| 6.8      | Experimental Results on Documents . . . . .          | 168       |

|                                                            |            |
|------------------------------------------------------------|------------|
| 6.8.1 Documents: Run-time Efficiency . . . . .             | 169        |
| 6.8.2 Documents: Quality and Imperceptibility . . . . .    | 170        |
| 6.8.3 Documents: Authenticity Verification . . . . .       | 171        |
| 6.8.4 Documents: Integrity Attestation . . . . .           | 172        |
| <b>7 Comparison with Related Work</b>                      | <b>174</b> |
| 7.1 Image Watermarking . . . . .                           | 175        |
| 7.2 Audio Watermarking . . . . .                           | 183        |
| 7.3 Video Watermarking . . . . .                           | 184        |
| 7.4 Documents Watermarking . . . . .                       | 189        |
| <b>8 Performance Analysis</b>                              | <b>192</b> |
| 8.1 Authenticity and Integrity Preservation . . . . .      | 192        |
| 8.1.1 Image . . . . .                                      | 192        |
| 8.1.2 Audio . . . . .                                      | 194        |
| 8.1.3 Video . . . . .                                      | 195        |
| 8.1.4 Document . . . . .                                   | 197        |
| 8.2 Practical Feasibility of the Proposed Models . . . . . | 199        |
| 8.3 Limitations and Challenges . . . . .                   | 201        |
| <b>9 Concluding Remarks</b>                                | <b>206</b> |
| 9.1 Research Results . . . . .                             | 206        |
| 9.2 Lessons Learned . . . . .                              | 207        |
| 9.3 Future Research . . . . .                              | 208        |
| <b>Bibliography</b>                                        | <b>210</b> |

# LIST OF FIGURES

---

|     |                                                                                                                                                                                                                                                                                                 |    |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1.1 | Categorization of Information Hiding Techniques based on watermark's visibility, sensitivity, payload and embedding domain . . . . .                                                                                                                                                            | 5  |
| 3.1 | Partition of a forensic image into authenticity compartments. . . . .                                                                                                                                                                                                                           | 28 |
| 3.2 | Partition of a medical image into integrity compartments. . . . .                                                                                                                                                                                                                               | 29 |
| 3.3 | Partition of an audio signal into the proposed compartments. . . . .                                                                                                                                                                                                                            | 30 |
| 3.4 | Partition video objects into image frames and corresponding audio sequence. . . . .                                                                                                                                                                                                             | 32 |
| 3.5 | Applying the forward and the inverse Arnold Transformation on a QR code. . . . .                                                                                                                                                                                                                | 35 |
| 3.6 | Encoding of an integer $W$ as a SiP. (a) Illustration of the encoding of an integer $W$ into a Self-inverting Permutation (SiP) and its corresponding mapping. (b) The resulting 2D representation of the SiP, where the * symbols denote the marked cells carrying the embedded information. . | 36 |
| 4.1 | Embedding and Extracting procedures utilizing Self-Inverting permutations forming the basis of our proposed image watermarking models.                                                                                                                                                          | 41 |
| 4.2 | Repetitive Embed procedure utilizing Self-Inverting Permutations. . . .                                                                                                                                                                                                                         | 44 |
| 4.3 | Repetitive Extract procedure utilizing Self-Inverting Permutations. . . .                                                                                                                                                                                                                       | 44 |
| 4.4 | Embed and Extract procedure of the image code-based multiple watermarking scheme. . . . .                                                                                                                                                                                                       | 46 |
| 4.5 | Arnold's transformation for the initial image (a) to develop its NFT "Dual" (b) for the ProtAIN model. . . . .                                                                                                                                                                                  | 56 |
| 4.6 | ProtAIN system architecture and procedure deployment. . . . .                                                                                                                                                                                                                                   | 56 |
| 4.7 | Overview of the procedures deployed for the embed and the extract of multiple watermarks into and from digital audio signals. . . . .                                                                                                                                                           | 58 |

|      |                                                                                                                                                                                                                                                                                                     |     |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 4.8  | Embedding procedure of the integer $w$ into the image $I_w$ followed by the embedding of $I_w$ into the LSB's of the audio signal $S$ . . . . .                                                                                                                                                     | 58  |
| 4.9  | Extracting procedure of the watermarked image $I_w$ from the LSB's of the audio signal $S_w$ followed by the extraction of the integer $w$ from $I_w$ . . . . .                                                                                                                                     | 61  |
| 4.10 | Overview of the video-audio binding process. Every video frame is processed by a cryptographic hash function to generate intermediate hash values, which are combined into a final hash. The final hash value is then embedded into the associated audio frames for integrity preservation. . . . . | 62  |
| 4.11 | Overview of the embedding of the final video-frame hash into selected audio samples through LSB modification. . . . .                                                                                                                                                                               | 65  |
| 4.12 | Architecture of the proposed video watermarking model. . . . .                                                                                                                                                                                                                                      | 66  |
| 4.13 | Example of the generated QR Code (a), the resized QR Code (b), the QR Code after the Arnold's transformation (c), the QR Code after the permutation rearrangement (d), and the extracted QR Code following the reverse procedure (e). . . . .                                                       | 69  |
| 4.14 | Example on the utilization of Arnold's transformation compared to the plain embedding of the QR Code's image against a targeted attack on the document's content. . . . .                                                                                                                           | 72  |
| 4.15 | Example of the QR Code reshaping and the embedding of the information using Arnold's transformation and permutation to achieve chaotic diffusion of the hidden information. . . . .                                                                                                                 | 72  |
| 4.16 | Integrity verification and tamper localization. (a) $\Lambda$ -relation among <i>Integrity-blocks</i> , illustrating verification dependencies. (b) Tamper detection and localization based on non-verifiable blocks. . . . .                                                                       | 78  |
| 6.1  | Example images from the general-purpose data-set utilized in the evaluation process, consisting of multiple image types. . . . .                                                                                                                                                                    | 98  |
| 6.2  | Example images from the medical image data-set utilized in the evaluation process, consisting of multiple image modalities. . . . .                                                                                                                                                                 | 99  |
| 6.3  | Example images from the forensic image data-set utilized in the evaluation process, consisting of multiple forensic scenes. . . . .                                                                                                                                                                 | 99  |
| 6.4  | Experimental data-set for the evaluation of the repetitive SiP model. . .                                                                                                                                                                                                                           | 118 |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                             |     |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 6.5  | Tamper detection and localization visualization results after evaluating our model for the preservation of integrity in forensic images . . . . .                                                                                                                                                                                                                                                                                                           | 133 |
| 6.6  | Tamper Detection and Localization under Different Attack Scenarios. Representative results across multiple resolution groups, illustrating insertion, removal, and combined attacks along with the corresponding detection and localization outputs. The highlighted regions indicate successfully identified tampered blocks, demonstrating the effectiveness of the proposed ROMFAIA system in accurately detecting and localizing modifications. . . . . | 135 |
| 6.7  | Crop attack demonstration for the evaluation of the dual-layer audio watermarking model. . . . .                                                                                                                                                                                                                                                                                                                                                            | 139 |
| 6.8  | Demonstration of the stable scalability and predictability in performance using the average per-frame run-time over different resolutions (720p, 1080p, 2K, and 4K) with frames-rates: 24, 30, 60 FPS, and H.264, HEVC and AV1 codecs for the code-based hash-chain video watermarking model. . . . .                                                                                                                                                       | 150 |
| 6.9  | Relationship between permutation size ( $P_s$ ) and embedding time ( $T_e$ ) for the evaluated CCTV videos. The observed positive linear trend indicates that the computational cost of the proposed framework increases proportionally to the number of elements participating in the cross-frame permutation process. . . . .                                                                                                                             | 152 |
| 6.10 | Visual quality comparison of the code-based hash chain video watermarking model between the original frame (a), the watermarked frame (b), and frames degraded by common signal processing and geometric attacks, including Gaussian blur ( $5 \times 5$ ) (c), Gaussian noise ( $\sigma = 0.4$ ) (d), histogram equalization (e), JPEG compression (Q=50) (f), and left-side cropping (50%) (g). . . . .                                                   | 155 |
| 6.11 | Example of Tampered Frames and Tamper Detection Results per frame under removal attack. . . . .                                                                                                                                                                                                                                                                                                                                                             | 165 |

|                                                                                                                                                                                                                                                                                                                                                                               |     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 6.12 Performance measurements during the embedding and the extraction phases of the PDF watermarking model showing respectively the required time during the embedding (a) and the extraction (b) phases, the CPU utilization during the embedding (c) and the extraction (d) phases, and the RAM utilization during the embedding (e) and the extraction (f) phases. . . . . | 169 |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|

# LIST OF TABLES

---

|     |                                                                                                                                                                                                                                                                                                                                 |     |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 4.1 | Security ablation evaluating the susceptibility of different frame–audio binding strategies to video-level manipulation under geometric, filtering, and audio tampering attacks. GIV stands for Geometric Integrity Vulnerability, FIV for Filtering Integrity Vulnerability and AIV for Audio Integrity Vulnerability. . . . . | 68  |
| 5.1 | Summary of the proposed authenticity and integrity preservation frameworks and their main architectural characteristics. . . . .                                                                                                                                                                                                | 91  |
| 6.1 | Characteristics of the VGG Sound video data-set . . . . .                                                                                                                                                                                                                                                                       | 100 |
| 6.2 | Characteristics of the CCTV video data-set . . . . .                                                                                                                                                                                                                                                                            | 101 |
| 6.3 | Embedding computational time of the proposed repetitive SiP watermarking model. . . . .                                                                                                                                                                                                                                         | 117 |
| 6.4 | Average execution times (in seconds) of the ROMFAIA watermarking model. . . . .                                                                                                                                                                                                                                                 | 119 |
| 6.5 | Imperceptibility measurements of the proposed repetitive SiP watermarking model. . . . .                                                                                                                                                                                                                                        | 121 |
| 6.6 | Imperceptibility evaluation of the image code-based multiple watermarking model. . . . .                                                                                                                                                                                                                                        | 122 |
| 6.7 | Average performance results per resolution group, reporting PSNR (dB), SSIM, Extraction-Rate $R_e$ (%), BER, and tamper detection outcomes, computed over all five images per resolution group for the ROMFAIA model. . . . .                                                                                                   | 123 |
| 6.8 | Imperceptibility evaluation of the image cross-block based on permutation watermarking model. . . . .                                                                                                                                                                                                                           | 124 |
| 6.9 | Extraction results of the repetitive SiP model in comparison with the basic SiP approach [1] under different cropping attacks. . . . .                                                                                                                                                                                          | 126 |



|      |                                                                                                                                                                                                                                                                                                                                                                                                                                            |     |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 6.10 | Extraction results of the repetitive model in comparison with the basic SiP approach [1] under JPEG compression attacks with different quality factors. . . . .                                                                                                                                                                                                                                                                            | 127 |
| 6.11 | Comparative results after crop and compression attacks. . . . .                                                                                                                                                                                                                                                                                                                                                                            | 128 |
| 6.12 | Code extraction rate $R_e$ (%) for different forensic images under various attacks for the evaluation of robustness of the code-based multiple watermarking model. . . . .                                                                                                                                                                                                                                                                 | 128 |
| 6.13 | Average robustness performance of the ROMFAIA model under cropping attacks. . . . .                                                                                                                                                                                                                                                                                                                                                        | 130 |
| 6.14 | Average robustness performance of the proposed ROMFAIA model under filtering and intensity-transformation attacks. . . . .                                                                                                                                                                                                                                                                                                                 | 130 |
| 6.15 | Robustness evaluation of the proposed ROMFAIA model under JPEG compression attacks. . . . .                                                                                                                                                                                                                                                                                                                                                | 131 |
| 6.16 | Average robustness performance of the proposed ROMFAIA model under geometric attacks. . . . .                                                                                                                                                                                                                                                                                                                                              | 132 |
| 6.17 | Experimental evaluation results of the proposed dual-layer audio watermarking model under crop, noise, and down-sampling attacks. . . .                                                                                                                                                                                                                                                                                                    | 141 |
| 6.18 | Tamper detection and localization results for the proposed permutation-based cross-audio block model under different attack categories. . . .                                                                                                                                                                                                                                                                                              | 144 |
| 6.19 | Evaluated attacks and corresponding parameters used for robustness assessment of the proposed code-based hash-chain video watermarking model. . . . .                                                                                                                                                                                                                                                                                      | 147 |
| 6.20 | Run-time benchmarking across different video resolutions. The time per frame for every resolution is the average of that resolution across all codecs and fps. The benchmark was conducted using parallelized execution of the embedding and extraction procedures utilizing 4 CPU cores. . . . .                                                                                                                                          | 149 |
| 6.21 | Average Embedding Time (ET) in seconds of the serialized and parallel implementations of the code-based hash-chain video watermarking model with corresponding approximate throughput in frames per second (FPS), and the achieved time reduction for different video resolutions. Runtime values are averaged across all tested frame rates and codecs at each resolution using a 4-core CPU. FPS is computed as $FPS = 1/Time$ . . . . . | 151 |

|      |                                                                                                                                                                                                                                                                                                                                                                                               |     |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 6.22 | Performance evaluation of the proposed cross-frame permutation-based video watermarking model in terms of perceptual quality and computational cost. . . . .                                                                                                                                                                                                                                  | 153 |
| 6.23 | PSNR and SSIM results of the code-based hash-chain video watermarking model across resolutions, codecs, and frame rates. . . . .                                                                                                                                                                                                                                                              | 154 |
| 6.24 | Verification and robustness results for the code-based hash-chain video watermarking model on 720p videos of H264, HEVC, and AV1 codecs across 24/30/60 FPS, depicting the mean $\pm$ standard deviation Extraction Rate( $R_e$ ) and BER for each row. . . . .                                                                                                                               | 158 |
| 6.25 | Verification and robustness results for the code-based hash-chain video watermarking model on 1080p videos of H264, HEVC, and AV1 codecs across 24/30/60 FPS, depicting the mean $\pm$ standard deviation Extraction Rate( $R_e$ ) and BER for each row. . . . .                                                                                                                              | 159 |
| 6.26 | Verification and robustness results for the code-based hash-chain video watermarking model on 2K videos of H264, HEVC, and AV1 codecs across 24/30/60 FPS, depicting the mean $\pm$ standard deviation Extraction Rate( $R_e$ ) and BER for each row. . . . .                                                                                                                                 | 160 |
| 6.27 | Verification and robustness results for the code-based hash-chain video watermarking model on 4K videos of H264, HEVC, and AV1 codecs across 24/30/60 FPS, depicting the mean $\pm$ standard deviation Extraction Rate( $R_e$ ) and BER for each row. . . . .                                                                                                                                 | 161 |
| 6.28 | Robustness and verification evaluation of the code-based hash-chain video watermarking model under spatial, temporal, compression, and combined transformations occurring after video sharing or transmission, i.e., Frame Rate Reduction (FRR), Frame Rate Increase (FRI), Spatial Downscale (SD), Spatial Upscale (SU), Re-Encoding (RE), Frame Dropping (FD), and Frame Swap (FS). . . . . | 163 |
| 6.29 | Average performance of the proposed cross-frame permutation based video watermarking model under single-frame tampering for different attack types. Metrics are averaged across all evaluated videos. . . . .                                                                                                                                                                                 | 166 |
| 6.30 | Average tamper detection and localization performance under multi-frame addition, removal and copy-move attacks for the cross-frame permutation based video watermarking model. . . . .                                                                                                                                                                                                       | 166 |

|      |                                                                                                                                                                                                                                                                       |     |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 6.31 | Imperceptibility metrics for all the PDF pages of the proposed PDF watermarking model. . . . .                                                                                                                                                                        | 170 |
| 6.32 | Quality, functionality, and authenticity-preservation results of the extracted QR Code under common PDF processing operations. . . . .                                                                                                                                | 171 |
| 6.33 | Quality, functionality, and authenticity-preservation results of the extracted QR Code after the performance of various content modification attacks. . . . .                                                                                                         | 172 |
| 7.1  | Comparison of watermark embedding computational time between the proposed model and the SIP-based watermarking technique [1]. . . . .                                                                                                                                 | 175 |
| 7.2  | Comparison of PSNR and SSIM measurements between the SIP-based watermarking technique [1] and the proposed model. . . . .                                                                                                                                             | 176 |
| 7.3  | Comparison of PSNR values between the proposed repetitive SiP watermarking model and representative image watermarking approaches from the literature. . . . .                                                                                                        | 177 |
| 7.4  | Comparison of computational performance in terms of average embedding ( $T^{Em}$ ) and extraction ( $T^{Ex}$ ) times of the ROMFAIA model. . . . .                                                                                                                    | 179 |
| 7.5  | Comparative imperceptibility results in terms of PSNR, SSIM, and BER under standard (no-attack) conditions of the ROMFAIA model. . . . .                                                                                                                              | 180 |
| 7.6  | Comparison of BER values reported in the literature under common attack scenarios for the ROMFAIA model. . . . .                                                                                                                                                      | 181 |
| 7.7  | Comparison of tamper detection performance in terms of Precision, Recall, F1-score, and Accuracy of the ROMFAIA model. . . . .                                                                                                                                        | 182 |
| 7.8  | Comparison of the audio dual-watermarking approach with recent literature approaches. . . . .                                                                                                                                                                         | 183 |
| 7.9  | Comparative performance (run-time) results with state-of-the-art methods including both traditional transform domain techniques and also Deep-Learning techniques, at resolution of $1280 \times 720$ for the code-based hash-chain video watermarking model. . . . . | 185 |
| 7.10 | Comparative imperceptibility results showcasing PSNR and SSIM in contrast to other approaches for the code-based hash-chain video watermarking model. . . . .                                                                                                         | 186 |

|      |                                                                                                                                                                                                                                                                                  |     |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 7.11 | Quantitative comparison based on reported BER values of representative video watermarking approaches under similar attack categories for the code-based hash-chain video watermarking model. . . . .                                                                             | 186 |
| 7.12 | Comparison in tamper detection with similar approaches (in chronological order) for the code-based hash-chain video watermarking model.                                                                                                                                          | 187 |
| 7.13 | Performance comparison with representative video tampering detection and localization approaches based on Precision, Recall, Accuracy, and F1-Score of the cross-frame permutation based video watermarking model. The metrics are averaged across all attack scenarios. . . . . | 189 |
| 7.14 | Comparison of the exhibited PSNR, SSIM and MSE values in regards to quality and imperceptibility of PDF pages with the results exhibited by related approaches. . . . .                                                                                                          | 190 |

# LIST OF ALGORITHMS

---

|      |                                                                                     |    |
|------|-------------------------------------------------------------------------------------|----|
| 4.1  | Repetitive Embedding Procedure . . . . .                                            | 42 |
| 4.2  | Repetitive Extraction Procedure . . . . .                                           | 43 |
| 4.3  | Code-based Multiple Embedding Procedure . . . . .                                   | 45 |
| 4.4  | Code-based Multiple Extraction Procedure . . . . .                                  | 47 |
| 4.5  | Image Authenticity and Integrity Embedding for the ROMFAIA Model                    | 49 |
| 4.6  | Image Authenticity and Integrity Verification for the ROMFAIA Model                 | 51 |
| 4.7  | Embed Watermarked Image to Audio Procedure . . . . .                                | 59 |
| 4.8  | Extract Watermarked Image from Audio Procedure . . . . .                            | 60 |
| 4.9  | Video Code-based Hash-Chain Embedding Algorithm . . . . .                           | 64 |
| 4.10 | Video Code-based Hash-Chain Extracting Algorithm . . . . .                          | 67 |
| 4.11 | Embed QR to PDF Algorithm . . . . .                                                 | 71 |
| 4.12 | Extract QR from PDF Algorithm . . . . .                                             | 74 |
| 4.13 | Cross-block Integrity Information Embedding Algorithm . . . . .                     | 76 |
| 4.14 | Tamper Detection and Localization based on $\Lambda$ -Relations Algorithm . .       | 77 |
| 4.15 | Embedding Integrity Information into Audio . . . . .                                | 81 |
| 4.16 | Audio Integrity Information Embedding - Secret Key Generation . . . .               | 82 |
| 4.17 | Tamper Detection and Localization in Audio . . . . .                                | 84 |
| 4.18 | Embedding Cross-modal Integrity Information in Visual and Audio<br>Frames . . . . . | 86 |
| 4.19 | Cross-modal Tamper Detection and Localization in Video . . . . .                    | 87 |

# ABSTRACT

---

Vasileios Vouronikos, Ph.D., Department of Computer Science and Engineering, School of Engineering, University of Ioannina, Greece, 2026.

Algorithmic Techniques for Information Hiding and Digital Object Protection.

Advisor: Stavros D. Nikolopoulos, Full Professor.

The widespread adoption of digital technologies and the continuous growth of multimedia content distribution have significantly increased the need for effective mechanisms capable of protecting digital objects against unauthorized modification, forgery, ownership violations, and content manipulation. Images, audio recordings, videos, and electronic documents are extensively utilized in domains such as digital forensics, healthcare, surveillance, and electronic communication, where authenticity and integrity are of critical importance. Consequently, the development of efficient information hiding and watermarking techniques for digital object protection remains an important research challenge.

For authenticity preservation, a collection of robust watermarking models is developed for images, audio signals, videos, and PDF documents. For images, watermark information is initially represented as integer sequences and encoded through Self-Inverting Permutations (SiPs), which are subsequently embedded in the frequency domain. The first proposed model employs repetitive embedding of a single SiP watermark across multiple image regions to improve robustness against cropping and partial-content attacks. Building upon this concept, a code-based multiple watermarking framework is introduced, where collections of integers are encoded as multiple SiPs and embedded simultaneously, increasing both watermark capacity and security while maintaining robustness against common signal-processing and geometric attacks. These contributions are further unified within the ROMFAIA framework, which combines robust authenticity preservation with fragile integrity verification for forensic medical images, and are subsequently extended to blockchain-based dig-

ital asset protection through the ProtAIN framework for secure NFT deployment. For audio signals, a dual-layer watermarking methodology is introduced by combining image watermarking and audio least significant bit embedding. For videos, a code-based hash-chain watermarking framework is proposed, combining frequency-domain watermarking with cryptographic frame-audio verification. For PDF documents, dynamically generated QR codes together with scrambling and permutation mechanisms are employed to support authenticity verification while preserving document readability and visual quality.

For integrity preservation, fragile watermarking techniques are proposed for images, audio signals, and videos. For images, a cryptographic integrity-verification framework is introduced based on inter-block verification relationships, enabling accurate tamper detection and localization of manipulated image areas while maintaining high visual quality. For audio signals, integrity preservation is achieved through the deployment of hash-based verification information embedded within audio samples, allowing the identification and localization of modified audio segments. For videos, two complementary integrity-preservation frameworks are developed. The first employs cryptographic hash chains embedded across video frames and synchronized with the audio stream, while the second introduces a cross-frame permutation framework capable of detecting addition, deletion, and copy-move manipulations. The proposed methodologies support reliable content verification and localization of modified regions while maintaining low computational complexity.

The proposed models are assessed in terms of imperceptibility, robustness, computational efficiency, and tamper detection performance using metrics including PSNR, SSIM, BER, NC, Extraction Rate, Precision, Recall, F1-score, Accuracy, and execution time. The obtained results demonstrate high visual quality, strong robustness against common attacks, efficient computational performance, and accurate tamper detection and localization capabilities. Comparative evaluations with representative approaches from the literature further confirm the effectiveness and competitiveness of the proposed methodologies.

Overall, this dissertation contributes a unified collection of watermarking and information hiding techniques for authenticity and integrity preservation in digital media. By combining frequency-domain and spatial-domain embedding strategies with combinatorial encoding structures, cryptographic verification mechanisms, and blockchain technologies, the proposed approaches provide effective and scalable so-

lutions for the protection of images, audio signals, videos, electronic documents, and NFT-based digital assets against forgery, manipulation, and unauthorized modification.



# ΕΚΤΕΤΑΜΕΝΗ ΠΕΡΙΛΗΨΗ

---

Βασίλειος Βουρονίκος, Δ.Δ., Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πολυτεχνική Σχολή, Πανεπιστήμιο Ιωαννίνων, 2026.

Αλγοριθμικές Τεχνικές Απόκρυψης Πληροφορίας και Προστασίας Ψηφιακών Αντικειμένων.

Επιβλέπων: Σταύρος Δ. Νικολόπουλος, Καθηγητής.

Η ραγδαία ανάπτυξη των ψηφιακών τεχνολογιών και η συνεχής αύξηση της ανταλλαγής πολυμεσικού περιεχομένου μέσω του Διαδικτύου έχουν δημιουργήσει αυξημένες απαιτήσεις για την προστασία της αυθεντικότητας και της ακεραιότητας των ψηφιακών αντικειμένων. Ψηφιακές εικόνες, ηχητικά αρχεία, βίντεο και ηλεκτρονικά έγγραφα χρησιμοποιούνται καθημερινά σε κρίσιμους τομείς, όπως η υγειονομική περίθαλψη, η ψηφιακή εγκληματολογία, η επιτήρηση, η ηλεκτρονική διακυβέρνηση και οι διαδικτυακές υπηρεσίες. Η δυνατότητα εύκολης επεξεργασίας και παραποίησης του περιεχομένου αυτών των αντικειμένων καθιστά αναγκαία την ανάπτυξη αποτελεσματικών τεχνικών προστασίας που να επιτρέπουν τόσο την επαλήθευση της αυθεντικότητας όσο και τον εντοπισμό μη εξουσιοδοτημένων τροποποιήσεων.

Αντικείμενο της παρούσας διδακτορικής διατριβής αποτελεί η μελέτη και ανάπτυξη αλγοριθμικών τεχνικών απόκρυψης πληροφορίας και ψηφιακής υδατογράφησης με στόχο τη διατήρηση της αυθεντικότητας και της ακεραιότητας ψηφιακών αντικειμένων. Η έρευνα εστιάζει στην ανάπτυξη τόσο ανθεκτικών (*robust*) όσο και εύθραυστων (*fragile*) μοντέλων υδατογράφησης, αξιοποιώντας τεχνικές του χωρικού και του συχνοτικού πεδίου, κρυπτογραφικούς μηχανισμούς, συνδυαστικές δομές και μηχανισμούς διασταυρούμενης επαλήθευσης πληροφορίας.

Στο πεδίο της διατήρησης της αυθεντικότητας ψηφιακών εικόνων προτείνονται νέα μοντέλα υδατογράφησης βασισμένα στις Αυτοαντιστρέφουσες Μεταθέσεις (Self-Inverting Permutations – SiPs). Αρχικά αναπτύσσεται ένα επαναληπτικό μοντέλο

υδατογράφησης, στο οποίο το ίδιο υδατογράφημα ενσωματώνεται πολλαπλές φορές σε διαφορετικές περιοχές της εικόνας με στόχο την ενίσχυση της ανθεκτικότητας έναντι επιθέσεων αποκοπής (*cropping*). Στη συνέχεια προτείνεται ένα κωδικοβασισμένο μοντέλο πολλαπλής υδατογράφησης, στο οποίο ακολουθίες ακεραίων κωδικοποιούνται σε πολλαπλές SiPs και ενσωματώνονται ταυτόχρονα στο συχνοτικό πεδίο της εικόνας, αυξάνοντας σημαντικά τόσο τη χωρητικότητα όσο και την ασφάλεια του συστήματος. Οι παραπάνω τεχνικές ενοποιούνται στο πλαίσιο του μοντέλου ROMFAIA, το οποίο συνδυάζει μηχανισμούς διατήρησης αυθεντικότητας και ακεραιότητας για ιατρικές εικόνες, ενώ επεκτείνονται περαιτέρω μέσω του μοντέλου ProtAIN για την προστασία ψηφιακών περιουσιακών στοιχείων τύπου NFT με αξιοποίηση τεχνολογιών blockchain και IPFS.

Για τη διατήρηση της ακεραιότητας ψηφιακών εικόνων προτείνεται ένα εύθραυστο μοντέλο υδατογράφησης βασισμένο σε κρυπτογραφικές συναρτήσεις SHA και σχέσεις διασταυρούμενης επαλήθευσης μεταξύ περιοχών της εικόνας. Η εικόνα διαμερίζεται σε περιοχές επαλήθευσης, ενώ η πληροφορία ακεραιότητας παράγεται και ενσωματώνεται σε διαφορετικές περιοχές σύμφωνα με αντιμεταθετικές αντιστοιχίσεις. Με τον τρόπο αυτό επιτυγχάνεται όχι μόνο η ανίχνευση αλλά και η ακριβής τοπικοποίηση των αλλοιωμένων περιοχών.

Στον τομέα του ψηφιακού ήχου αναπτύσσονται μοντέλα αυθεντικότητας και ακεραιότητας τα οποία συνδυάζουν τεχνικές υδατογράφησης εικόνων με μηχανισμούς ενσωμάτωσης στο λιγότερο σημαντικό δυαδικό ψηφίο (Least Significant Bit – LSB) των ηχητικών δειγμάτων. Τα προτεινόμενα συστήματα επιτρέπουν την ενσωμάτωση πληροφορίας επαλήθευσης και την αξιόπιστη ανίχνευση και τοπικοποίηση μη εξουσιοδοτημένων μεταβολών του ηχητικού περιεχομένου.

Στον τομέα του ψηφιακού βίντεο αναπτύσσονται δύο συμπληρωματικά μοντέλα διατήρησης της αυθεντικότητας και της ακεραιότητας. Το πρώτο αποτελεί ένα κωδικοβασισμένο μοντέλο διατήρησης αυθεντικότητας, στο οποίο η πληροφορία αυθεντικότητας ενσωματώνεται στα καρέ του βίντεο, ενώ μία κρυπτογραφική αναπαράσταση όλων των καρέ ενσωματώνεται συγχρονισμένα στο ηχητικό κανάλι, δημιουργώντας ισχυρή σύνδεση μεταξύ οπτικού και ακουστικού περιεχομένου. Η δεύτερη προσέγγιση αφορά ένα μοντέλο διατήρησης ακεραιότητας βασισμένο σε μεταθέσεις μεταξύ μπλοκ διαφορετικών καρέ (*cross-frame permutations*), το οποίο επιτρέπει την ανίχνευση και τοπικοποίηση αλλοιώσεων τόσο σε επίπεδο αντικειμένου όσο και σε επίπεδο καρέ. Ειδικότερα, το προτεινόμενο σύστημα είναι ικανό να ανιχνεύει επιθέ-

σεις εισαγωγής, διαγραφής και αντιγραφής-μετακίνησης αντικειμένων (*copy-move*), καθώς και επιθέσεις εισαγωγής, διπλασιασμού και διαγραφής καρέ (*frame insertion, frame duplication και frame deletion*), παρέχοντας παράλληλα δυνατότητα ακριβούς τοπικοποίησης των αλλοιωμένων περιοχών.

Για τα ηλεκτρονικά έγγραφα PDF προτείνεται μία νέα τεχνική υδατογράφησης βασισμένη σε δυναμικά παραγόμενους κώδικες QR. Η πληροφορία υδατογράφησης μετασχηματίζεται μέσω μετασχηματισμού Arnold και τεχνικών αναδιάταξης πριν ενσωματωθεί στα χρωματικά χαρακτηριστικά επιλεγμένων χαρακτήρων του εγγράφου. Η προσέγγιση αυτή επιτρέπει την επαλήθευση της αυθεντικότητας χωρίς να επηρεάζεται η αναγνωσιμότητα, η λειτουργικότητα ή η οπτική ποιότητα του εγγράφου.

Για την αξιολόγηση των προτεινόμενων μοντέλων πραγματοποιήθηκε εκτεταμένη πειραματική μελέτη σε σύνολα δεδομένων εικόνων, ιατρικών εικόνων, εγκληματολογικών εικόνων, ηχητικών αρχείων, βίντεο και εγγράφων PDF. Η αξιολόγηση πραγματοποιήθηκε με τη χρήση μετρικών όπως PSNR, SSIM, BER, NC, Precision, Recall, F1-score, Accuracy και χρόνου εκτέλεσης. Τα πειραματικά αποτελέσματα καταδεικνύουν ότι τα προτεινόμενα συστήματα επιτυγχάνουν υψηλή οπτική ποιότητα, ισχυρή ανθεκτικότητα έναντι κοινών επιθέσεων επεξεργασίας σήματος, αποδοτική υπολογιστική συμπεριφορά και ιδιαίτερα υψηλή ακρίβεια στην ανίχνευση και τοπικοποίηση αλλοιώσεων.

Συνολικά, η παρούσα διατριβή συνεισφέρει μία ολοκληρωμένη οικογένεια τεχνικών απόκρυψης πληροφορίας και ψηφιακής υδατογράφησης για την προστασία της αυθεντικότητας και της ακεραιότητας ψηφιακών αντικειμένων. Οι προτεινόμενες μέθοδοι συνδυάζουν τεχνικές του χωρικού και του συχνοτικού πεδίου, κρυπτογραφικούς μηχανισμούς, συνδυαστικές δομές και τεχνολογίες blockchain, παρέχοντας αποτελεσματικές και επεκτάσιμες λύσεις για την προστασία εικόνων, ήχου, βίντεο, ηλεκτρονικών εγγράφων και ψηφιακών περιουσιακών στοιχείων τύπου NFT από παραποίηση, αλλοίωση και μη εξουσιοδοτημένες τροποποιήσεις.

# CHAPTER 1

## INTRODUCTION

- 
- 1.1 Digital Object Types**
  - 1.2 Digital Object Security**
  - 1.3 Information Hiding**
  - 1.4 Authenticity and Integrity Preservation**
  - 1.5 Thesis Contributions**
  - 1.6 Structure of the Thesis**
- 

### 1.1 Digital Object Types

The rapid increase in information exchange through the Internet, along with the digitization of many sectors of everyday life, has led to a significant increase in the usage of different types of digital objects. Digital objects may vary from simple digital files to complex datasets that combine both metadata and content. Among the most widely used digital objects in our everyday life are digital images, videos, audio files, and digital documents such as PDF files.

Digital images are extensively utilized across various social media platforms, enabling users to publicly share and distribute visual content over the Internet. Moreover, digital images are frequently utilized for commercial and profit-oriented purposes, including digital art assets such as NFTs, professional photography galleries, and visual assets used in video games. Similarly, digital videos constitute one of the

dominant forms of online content distribution, widely adopted in entertainment platforms, surveillance systems, streaming services, and social media applications. Audio files are also extensively used in communication, music streaming, and multimedia applications, while digital documents such as PDF files are commonly employed for the storage and exchange of official, academic, and legal information.

The ease with which digital objects can be exchanged in everyday life has significantly improved information accessibility and communication. However, despite the numerous advantages offered by digital media, their widespread distribution also introduces several threats. Unauthorized modifications may be performed by virtually any user through widely accessible and often freely available software tools and applications, with the intention of falsely claiming ownership of media content, altering evidence, or spread misleading information through forged or manipulated data. In recent years, numerous incidents have demonstrated for example the misuse of artificial intelligence technologies in order to generate fabricated images, videos, and audio content intended to mislead users of online and social media platforms.

Consequently, the protection of digital content against unauthorized modification, forgery, and ownership violations has become a critical research challenge. In this context, significant research attention has been directed toward the development of techniques capable of ensuring the authenticity, integrity, and copyright protection of digital objects. Among these techniques, digital watermarking has emerged as one of the most widely adopted approaches for embedding imperceptible information into digital media for authentication, ownership verification, and tamper detection purposes.

## **1.2 Digital Object Security**

Digital object security refers to the set of algorithms, techniques, and methods designed to protect the authenticity and integrity of digital objects against malicious manipulations and risks arising from content distribution and exchange. Digital object security has become increasingly important with the widespread availability of advanced content generation and editing tools. Recent advances in Artificial Intelligence, particularly Generative Adversarial Networks (GANs) and other generative models, have enabled the creation of highly realistic synthetic images, audio record-

ings, videos, and textual content. Such technologies are capable of producing content that closely resembles authentic data, making it increasingly difficult to distinguish genuine digital objects from manipulated or AI-generated media. The rapid proliferation of synthetic media and deepfake technologies has raised significant concerns regarding misinformation, digital forgery, privacy violations, and the overall trustworthiness of digital information. Consequently, the development of reliable authenticity and integrity preservation mechanisms has become a fundamental requirement for modern multimedia systems and digital communication environments [2]. To address the challenges associated with digital object security, a number of fundamental security requirements must be satisfied. These requirements aim to ensure the authenticity, integrity, reliability, and trustworthiness of digital content throughout its lifecycle. The most important threats and requirements are presented below:

- **Unauthorized Content Modification:** Digital objects can be altered through content insertion, deletion, replacement, or manipulation, potentially changing their meaning and reducing their trustworthiness.
- **Ownership Verification:** The widespread distribution of digital content increases the need for mechanisms capable of verifying the legitimate owner or source of a digital object.
- **Tamper Detection and Localization:** Security systems should not only identify whether a digital object has been modified, but also determine the location and extent of the modifications.
- **Resistance to Signal-Processing Attacks:** Digital objects frequently undergo operations such as compression, filtering, scaling, and format conversion. Security mechanisms should remain effective under such transformations.
- **Protection Against Forgery:** Modern editing tools enable the creation of convincing forged content, making the detection of manipulated digital objects an important security requirement.
- **Trustworthy Multimedia Distribution:** The protection of authenticity and integrity is essential in domains such as digital forensics, healthcare, surveillance, journalism, and electronic document management, where decisions are often based on the reliability of digital evidence.

The protection of a digital object's authenticity against malicious manipulations ensures that its ownership can be verified and that the content can be distributed without compromising its reliability and trustworthiness. On the other hand, the deployment of algorithms designed to protect the integrity of digital objects enables the detection and localization of malicious modifications whenever they occur. This capability is particularly important in domains such as forensic analysis and medical systems, where the authenticity and reliability of digital data are of critical importance.

### **1.3 Information Hiding**

Information hiding refers to the process of embedding data into a host digital object in such a way that the existence of the hidden information remains imperceptible. Information hiding can be broadly divided into two main categories: steganography and digital watermarking. Steganographic techniques primarily focus on concealing the existence of a hidden message so that it cannot be perceived by unauthorized parties during the communication process. In steganography, the hidden message itself constitutes the valuable information, while the host digital object mainly serves as a transmission medium. In contrast, digital watermarking techniques are primarily designed to protect the authenticity, integrity, and ownership of digital objects through the embedding of unique identifiers into the host digital content. A fundamental requirement in information hiding is that the embedded information should remain imperceptible within the carrier digital object. Once the hidden information becomes detectable by an unauthorized recipient, recovering the concealed message may eventually become possible. Information hiding is also frequently confused with cryptography, although the two concepts are fundamentally different. In cryptography, a message is transformed into an unintelligible form and typically requires a decryption key in order to be restored to its original state.

Initially information hiding techniques can be categorized according to their resilience against attacks. According to [3] and [4] information hiding schemes can be classified as fragile, semi-fragile, and robust. Fragile techniques are highly sensitive to modifications, causing the watermark to be easily destroyed even under minor alterations. Semi-fragile techniques tolerate distortions only up to a predefined robustness threshold; once this threshold is exceeded, the watermark becomes corrupted. Fi-

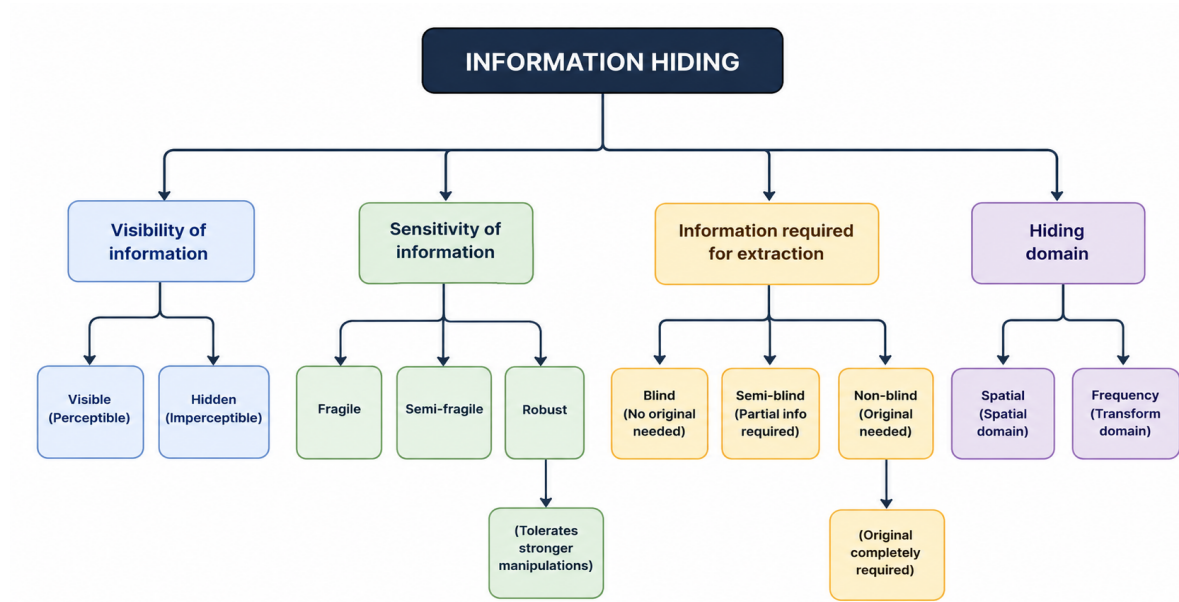


Figure 1.1: Categorization of Information Hiding Techniques based on watermark's visibility, sensitivity, payload and embedding domain

nally, robust techniques are designed to withstand a wide range of attacks, allowing the watermark to remain detectable even after severe distortions that may significantly degrade the quality of the digital object.

Information hiding techniques can be also categorized based on the domain in which the hidden information is embedded to spatial-domain techniques and frequency-domain techniques. In frequency domain the content of the digital object is transformed from the spatial domain into its corresponding frequency domain representation. For example, in the case of digital images, the pixel values are transformed to frequency coefficients. On the other hand spatial domain techniques operate directly on the original values of the digital object without transforming it into another representation. For example, in PDF files, the embedding process is performed on the colors of the text characters or on the structure of the PDF. Frequency domain techniques utilize frequency transformations including Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) and others. Spatial domain techniques can also use spatial transformations like Singular Value Decomposition (SVD) or bit value substitutions like LSB replacement.

Another important categorization of information hiding techniques concerns the amount of information required for watermark extraction. According to [3], watermarking schemes can be divided into three categories based on the information



needed during the extraction process: blind, semi-blind, and non-blind watermarking. In techniques using blind embedding approaches, the extraction process does not require any information from the original signal. However, such techniques often exhibit lower robustness against attacks. Semi-blind techniques require partial information from the original signal in order to recover the watermark. Finally, non-blind approaches require access to the original signal for successful watermark extraction.

Next we describe the fundamental performance requirements and characteristics that information hiding techniques must satisfy. More precisely:

- **Imperceptibility and Quality:** Imperceptibility refers to the ability of an information hiding technique to embed information without introducing noticeable visual or perceptual distortions. It is commonly evaluated using metrics such as PSNR and SSIM, which quantify the distortion introduced by the embedding process and the similarity between the original and the watermarked digital object. High imperceptibility is particularly important in applications requiring high-quality content, such as medical and forensic systems.
- **Robustness and Fragility:** Information hiding techniques can be classified as robust or fragile depending on their intended application. Robust techniques are designed to withstand common signal-processing and geometric attacks while preserving the embedded information, whereas fragile techniques are highly sensitive to modifications and are primarily used for integrity verification, tamper detection, and localization. Their performance is typically evaluated through watermark extraction accuracy or tamper detection metrics.
- **Payload:** Payload denotes the amount of information that can be embedded into a digital object. Although increased payload capacity enables the storage of additional information, it often introduces greater distortion. Consequently, watermarking systems must balance payload capacity with robustness and imperceptibility requirements.
- **Security:** Security refers to the ability of a watermarking system to prevent unauthorized users from detecting, modifying, or extracting the embedded information. This objective is commonly achieved through the use of secret keys, cryptographic primitives, and secure embedding procedures.

- **Computational Complexity:** Computational complexity describes the resources required for watermark embedding and extraction. While frequency-domain techniques generally provide improved robustness, they often require higher computational cost than spatial-domain approaches due to the transformations involved in the embedding process.

## 1.4 Authenticity and Integrity Preservation

Authenticity and integrity preservation constitute two of the most important objectives of information hiding and digital watermarking systems. Although the two concepts are closely related, they address different security requirements. Authenticity preservation focuses on verifying the legitimate ownership or origin of a digital object, ensuring that ownership information can be reliably recovered even after the object has undergone signal-processing operations or malicious attacks. Integrity preservation, on the other hand, aims to detect unauthorized modifications and determine whether the content of a digital object has been altered.

Authenticity preservation is commonly achieved through robust watermarking techniques. In such systems, watermark information is embedded into the host digital object in a manner that allows successful extraction even after the application of common attacks, including compression, filtering, noise addition, geometric transformations, and format conversion. The primary objective of these techniques is to maintain watermark recoverability while preserving the perceptual quality of the host object.

In contrast, integrity preservation is typically achieved through fragile watermarking techniques. Unlike robust watermarking, fragile watermarking schemes are intentionally designed to be sensitive to modifications. Consequently, any unauthorized alteration of the watermarked content affects the embedded information and can be detected during the verification process. In many cases, fragile watermarking systems additionally provide tamper localization capabilities, enabling the identification of the manipulated regions within the digital object.

The design of authenticity and integrity preservation systems presents several challenges. Robust watermarking techniques must achieve a suitable balance between robustness, imperceptibility, payload capacity, and security, while fragile watermarking

techniques must provide accurate tamper detection and localization without generating excessive false detections. Furthermore, modern multimedia applications often require these properties to be maintained across different types of digital objects, including images, audio signals, videos, and electronic documents.

In this dissertation, both authenticity-preservation and integrity-preservation methodologies are investigated. Robust watermarking frameworks are proposed for images, audio signals, videos, and PDF documents to support ownership verification and authenticity validation. In parallel, fragile watermarking techniques based on cryptographic verification mechanisms are developed for images, audio signals, and videos, enabling reliable tamper detection and localization. Together, these approaches provide a comprehensive suite of frameworks for protecting digital objects against forgery, manipulation, and unauthorized modification.

## 1.5 Thesis Contributions

This dissertation's research focuses on the design of robust and fragile watermarking models for the protection of authenticity and integrity of images, audio signals, videos, and PDF documents. The proposed methodologies combine frequency-domain and spatial-domain embedding techniques with combinatorial structures, cryptographic techniques, and verification mechanisms in order to improve robustness, security, computational efficiency, and tamper detection capabilities. The main contributions of this dissertation can be summarized as follows:

- The introduction of a repetitive SiP embedding strategy, enabling redundant watermark placement across multiple regions of a digital object and improving robustness against cropping and partial-content attacks.
- The development of a code-based watermark representation model, where multiple integers are encoded as a collection of SiPs that collectively represent a unique watermark code. Unlike traditional SiP-based approaches that encode a single integer, the proposed model supports higher-capacity watermark representations while improving security and authenticity verification capabilities.
- The introduction of a deterministic image integrity-preservation model based on cryptographic hashes and inter-block verification relationships, enabling reliable

tamper detection and localization without relying on probabilistic or learning-based decision mechanisms.

- The development of the ROMFAIA model, a unified architecture that combines robust authenticity preservation and fragile integrity verification for forensic medical images. The framework further enhances the security of the proposed integrity-preservation mechanism by replacing SHA-256 with HMAC-SHA-256, introducing secret-key authentication that prevents the unauthorized regeneration of valid verification information while preserving high imperceptibility and reliable tamper localization.
- The extension of the ROMFAIA authenticity and integrity preservation model to the domain of Non-Fungible Tokens through the ProtAIN framework, introducing automated smart contract generation and an NFT “dual” backup mechanism based on Arnold transformations, enabling future-proof attestation of NFT image authenticity and integrity independent of marketplace or platform trust.
- The introduction of a dual-layer audio watermarking architecture that utilizes a frequency-domain SiP-watermarked image as the embedded payload. Unlike conventional approaches that directly embed logos or images, the proposed model embeds a robust watermark carrier, improving the recoverability and authenticity verification capabilities of the watermarked audio signal.
- The introduction of an audio integrity-preservation mechanism in which authentication information is generated through XOR operations between audio-block data and HMAC-derived cryptographic keys, and subsequently embedded into different audio blocks. This design enables reliable and deterministic tamper detection and localization.
- The development of a global frame–audio binding mechanism for video authenticity preservation, where a cryptographic integrity representation derived from all video frames is embedded into the audio stream. This design establishes interdependencies between the visual and auditory components of the video, strengthening authenticity verification and improving resistance against spatial, temporal, and audio tampering attacks.

- The introduction of a video integrity-preservation framework based on cross-frame permutations and cross-block authentication relationships, where verification information is additionally bound to the audio stream. This multi-layer verification structure enables reliable detection and localization of object insertion, object deletion and copy-move attacks.
- The introduction of a QR-code-based PDF watermarking framework that combines Arnold transformations with an additional permutation layer to conceal characteristic QR-code patterns and prevent attackers from identifying watermark locations. Watermark information is embedded through color modifications of text objects, avoiding document rasterization and preserving the readability, functionality, and visual quality of the original PDF document.
- An extensive experimental evaluation framework for authenticity and integrity preservation systems, covering images, medical images, audio signals, videos, and PDF documents under a wide range of attack scenarios.

Overall, the dissertation contributes a comprehensive collection of information hiding and digital watermarking techniques for authenticity and integrity preservation across multiple categories of digital objects, including images, audio signals, videos, and electronic documents.

## 1.6 Structure of the Thesis

This dissertation is organized into nine chapters. Chapter 1 introduces the research area and presents the fundamental concepts of digital objects, digital object security, information hiding, authenticity preservation, integrity preservation, and the main contributions of the dissertation. Next, Chapter 2 reviews the theoretical background and the related work on information hiding, digital watermarking, and authenticity and integrity preservation systems. Subsequently, Chapter 3 presents the fundamental components, digital object characteristics, segmentation rules, watermarking domains, and cryptographic mechanisms utilized throughout the proposed methodologies. Building upon this foundation, Chapter 4 introduces the proposed authenticity and integrity preservation models for images, audio signals, videos, and PDF documents. Chapter 5 then presents the software architecture, implementation details,

and integration procedures of the developed frameworks. Consequently, Chapter 6 describes the experimental evaluation methodology, including the utilized data-sets, performance metrics, attack scenarios, and the obtained experimental results. Following the experimental assessment, Chapter 7 provides a comparative analysis between the proposed methodologies and representative approaches from the literature. Next, Chapter 8 discusses the overall performance of the proposed frameworks, focusing on their authenticity and integrity preservation capabilities, practical feasibility, limitations, and remaining challenges. Finally, Chapter 9 summarizes the main findings of the dissertation, highlights the lessons learned during the conducted research, and outlines potential directions for future work.

## CHAPTER 2

# THEORETICAL BACKGROUND

---

### 2.1 Related Work on Information Hiding and Watermarking

### 2.2 Information Hiding in the Frequency Domain

### 2.3 Information Hiding in the Spatial Domain

### 2.4 Related Approaches in Authenticity and Integrity Preservation

---

## 2.1 Related Work on Information Hiding and Watermarking

Information hiding and digital watermarking constitute fundamental research areas in multimedia security, aiming to embed information within digital media while preserving the perceptual quality of the host object. Depending on the application requirements, embedded information may be utilized for copyright protection, ownership verification, authentication, integrity preservation, tamper detection, content tracking, or secure information transmission. Over the past decades, a significant number of watermarking techniques have been proposed for images, audio, video, and document files, employing a wide variety of embedding strategies and signal processing methodologies.

The effectiveness of a watermarking scheme is commonly evaluated according to three principal criteria. Imperceptibility refers to the ability of the embedded watermark to remain visually or perceptually unnoticeable. Robustness measures the capability of the watermark to survive intentional and unintentional modifications,

including compression, filtering, noise addition, and geometric transformations. Capacity describes the amount of information that can be embedded while maintaining acceptable perceptual quality. Achieving a suitable balance among these often conflicting objectives remains one of the primary challenges in watermarking research.

Recent advances have significantly expanded the scope of information hiding beyond traditional copyright protection applications. Modern watermarking systems increasingly address authenticity verification, integrity preservation, tamper detection, and content authentication. In particular, fragile and semi-fragile watermarking techniques have been developed to identify unauthorized modifications, while robust watermarking schemes continue to focus on maintaining watermark recoverability under common signal processing attacks. Furthermore, recent studies [5, 6, 7, 8] illustrate this expanding scope across different media types, ranging from audio steganography [5] to robust and high-capacity video watermarking techniques designed to withstand transcoding, hybrid attacks, and large-scale content distribution [6, 7, 8].

As the field has evolved, watermarking approaches have generally followed three major directions. The first direction consists of frequency-domain techniques, which embed watermark information into transformed representations of multimedia objects. The second direction includes spatial-domain techniques, where information is embedded directly into the host signal. More recently, machine learning and deep learning methodologies have emerged as a third direction, utilizing data-driven models for adaptive watermark embedding and extraction. The following sections review these categories and discuss their applicability to authenticity and integrity preservation systems.

## **2.2 Information Hiding in the Frequency Domain**

Frequency-domain watermarking techniques constitute one of the most extensively investigated categories of information hiding approaches. Unlike spatial-domain methods, which directly modify pixel values, frequency-domain techniques first transform the host multimedia object into an alternative mathematical representation and subsequently embed watermark information into selected transform coefficients. Commonly employed transforms include the Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT), Singular Value Decomposi-



tion (SVD), and various combinations thereof. By embedding information within frequency coefficients rather than directly manipulating pixel intensities, these approaches generally achieve improved robustness against compression, filtering, noise addition, and other common signal-processing operations.

Among the earliest frequency-domain watermarking methodologies, DWT-based techniques have attracted significant attention due to their ability to provide multi-resolution representations of multimedia content. By decomposing an image into different frequency sub-bands, DWT enables watermark information to be embedded within regions that offer a favorable balance between imperceptibility and robustness. Several studies have exploited this property to develop watermarking systems capable of maintaining watermark recoverability under various image processing attacks. For example, Roy and Pal [9] combined DWT and SVD transformations to improve watermark robustness while preserving visual quality. Similarly, Jamal et al. [10] proposed a DWT-based watermarking framework that demonstrated resilience against common filtering and compression operations.

Singular Value Decomposition has also been widely employed within frequency-domain watermarking systems due to the stability of singular values under moderate image modifications. SVD-based techniques typically embed watermark information by modifying selected singular values while preserving the overall visual structure of the host image. Several researchers have combined SVD with other transforms, particularly DWT and DCT, to exploit the advantages of multiple frequency representations simultaneously. Such hybrid transform-domain schemes generally provide improved robustness compared to single-transform approaches while maintaining high imperceptibility.

Discrete Cosine Transform based watermarking constitutes another major category of frequency-domain information hiding techniques. DCT is particularly attractive because of its direct relationship to JPEG compression, allowing watermark information to be embedded within coefficients that remain relatively stable during compression. Ying et al. [11] proposed a robust DCT-based watermarking methodology designed to preserve watermark recoverability under lossy compression. Similarly, Begum et al. [12] utilized DCT coefficients to achieve a favorable balance between embedding capacity and visual quality. Other researchers have combined DCT with optimization and decomposition techniques to further improve watermark robustness and imperceptibility.

Fourier-based watermarking approaches represent another important family of frequency-domain techniques. The Discrete Fourier Transform provides desirable properties such as translation invariance and strong resistance to certain geometric attacks. Wang et al. [13] exploited the Quaternion Fourier Transform to enhance watermark embedding in color images while preserving perceptual quality and improving robustness against common processing operations. Chroni et al. [14] introduced a code-based watermarking framework in which watermark information is embedded directly within image regions using combinatorial structures.

## 2.3 Information Hiding in the Spatial Domain

Spatial-domain watermarking techniques embed information directly into the host multimedia object by modifying its original samples, pixels, or structural components without employing intermediate mathematical transformations. Due to their simplicity, low computational complexity, and ease of implementation, spatial-domain techniques have been extensively investigated for image, audio, video, and document watermarking applications. In contrast to frequency-domain approaches, which operate on transformed representations of multimedia content, spatial-domain methods directly manipulate the original signal, making them particularly attractive for real-time and resource-constrained environments.

Among the most widely utilized spatial-domain techniques are Least Significant Bit (LSB) based approaches. These methods embed watermark information by modifying the least significant bits of image pixels, audio samples, or video frames, resulting in minimal perceptual distortion. LSB watermarking is characterized by high embedding capacity and low computational requirements, making it suitable for applications where efficiency is prioritized. Several studies have explored variations of LSB embedding to improve robustness and security. For example, classical LSB schemes [15, 16] demonstrated the feasibility of imperceptible information embedding through direct bit substitution. Beyond traditional LSB techniques, numerous researchers have investigated block-based spatial watermarking methodologies. These approaches partition the host image into smaller regions and embed watermark information within selected blocks according to predefined embedding strategies. Block-based methods typically offer improved localization capabilities and can facilitate tamper detection

and authentication applications.

Document watermarking constitutes another important application area of spatial-domain information hiding. Unlike traditional image watermarking systems, document watermarking techniques frequently exploit structural properties of text objects, character formatting, spacing characteristics, and color information. Several researchers have proposed document authentication frameworks based on direct modifications of textual or graphical components. In particular, PDF watermarking methodologies [17, 18, 19] have demonstrated that watermark information can be embedded within document structures while preserving readability and visual appearance. Such approaches are particularly relevant for authenticity preservation applications, where unauthorized modifications must be detected without significantly altering the original document content. Spatial-domain techniques have also been extensively employed in audio watermarking applications. Rather than modifying transform coefficients, audio spatial watermarking schemes directly manipulate audio samples or selected sample bits to embed authentication or ownership information.

A significant advantage of spatial-domain watermarking techniques is their relatively low computational complexity. Since no transform-domain calculations are required, embedding and extraction procedures can often be performed significantly faster than corresponding frequency-domain approaches. Furthermore, spatial methods typically offer higher embedding capacities because watermark information is inserted directly into the host medium. These characteristics make spatial-domain techniques particularly attractive for real-time systems and applications involving large multimedia datasets.

## **2.4 Related Approaches in Authenticity and Integrity Preservation**

The increasing proliferation of digital multimedia content has significantly intensified concerns regarding authenticity, integrity, and trustworthiness. As digital images, videos, audio recordings, and electronic documents can be easily manipulated using modern editing tools, numerous watermarking and authentication techniques have been proposed to detect unauthorized modifications and verify content integrity. Unlike conventional copyright-oriented watermarking schemes, authenticity and integrity preservation approaches primarily focus on identifying whether content has

been altered and, in many cases, localizing the tampered regions.

A large category of authentication techniques relies on fragile and semi-fragile watermarking mechanisms. Fragile watermarking schemes are intentionally designed so that even minor modifications to the protected content result in corruption or destruction of the embedded watermark. Consequently, any discrepancy between the extracted and the original watermark indicates potential tampering. Semi-fragile watermarking techniques extend this concept by tolerating acceptable processing operations such as compression while remaining sensitive to malicious modifications. Such approaches have been extensively investigated for image authentication, medical image protection, document verification, and video tamper detection.

Medical imaging represents one of the most critical application domains for authenticity-preserving watermarking systems. In medical environments, unauthorized modifications can compromise diagnostic accuracy and potentially endanger patient safety. Qasim et al. [20] proposed a watermarking framework that exploits regions of interest and regions of non-interest to preserve diagnostic information while enabling integrity verification. Similarly, Mathew et al. [21] introduced a reversible watermarking scheme capable of restoring the original medical image after watermark extraction. Laishram et al. [22] further explored watermarking techniques for secure medical image authentication, demonstrating the importance of balancing imperceptibility, reversibility, and integrity preservation. Gull et al. [23] further proposed an efficient watermarking technique for tamper detection and localization in medical images, demonstrating the effectiveness of watermark-assisted verification for detecting content modifications while preserving acceptable visual quality.

In addition to watermark-based authentication, several researchers have incorporated cryptographic mechanisms into multimedia protection frameworks. Such approaches typically combine watermarking with cryptographic hashes, digital signatures, encryption procedures, or message authentication codes to strengthen integrity verification capabilities. Liu et al. [24] proposed an authentication framework that integrates watermarking with cryptographic verification procedures, while Sayah et al. [25] developed a secure watermarking methodology emphasizing both content protection and authenticity validation. More recently, Taj et al. [26] and Bouarroudj et al. [27] proposed lightweight authentication schemes designed to provide efficient integrity verification while maintaining low computational complexity.

The emergence of blockchain technology has also influenced recent developments

in authenticity preservation systems. Blockchain-based approaches utilize decentralized ledgers to record ownership information, authentication records, and watermark verification results. Such systems aim to provide immutable evidence of ownership and content authenticity. Blockchain-assisted watermarking schemes [28] demonstrate how distributed ledger technologies can complement traditional watermarking mechanisms by providing transparent and tamper-resistant verification infrastructures.

### **2.4.1 Frequency Based Watermarking Techniques**

Frequency-based watermarking techniques represent one of the most mature and widely adopted categories of digital watermarking methods. Unlike spatial-domain approaches that directly modify the original multimedia content, frequency-domain techniques first transform the host object into an alternative mathematical representation and subsequently embed watermark information within selected transform coefficients. Commonly employed transformations include the Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Discrete Fourier Transform (DFT), Singular Value Decomposition (SVD), and combinations thereof. The principal advantage of frequency-domain watermarking lies in its ability to achieve improved robustness against compression, filtering, noise addition, and various signal-processing attacks while maintaining acceptable perceptual quality.

Among the most extensively investigated transform-domain methodologies are DWT-based watermarking techniques. By decomposing an image into multiple frequency sub-bands, DWT enables watermark information to be embedded within regions that provide a favorable balance between imperceptibility and robustness. Sharma and Chandrasekaran [29] further demonstrated that combining multiple transform-domain representations can significantly improve watermark robustness against geometric distortions and compression operations. Singular Value Decomposition has frequently been integrated into frequency-domain watermarking systems due to the stability of singular values under moderate modifications. SVD-based techniques generally embed watermark information by modifying selected singular values of transformed image components, thereby preserving the structural characteristics of the host content. Several researchers have combined SVD with DWT and DCT representations to exploit the advantages of multiple transform domains simulta-

neously. Such approaches typically provide improved robustness while maintaining high imperceptibility.

Discrete Cosine Transform based watermarking techniques have also attracted significant research interest due to their compatibility with JPEG compression. By embedding watermark information within selected DCT coefficients, these methods can preserve watermark recoverability even after lossy compression. Other studies have combined DCT with optimization and decomposition techniques to further enhance watermark resilience and imperceptibility.

Fourier-domain watermarking constitutes another important category of frequency-based information hiding. The Discrete Fourier Transform offers desirable properties such as translation invariance and robustness against certain geometric transformations. Chroni et al. [30] introduced a code-based watermarking scheme that embeds combinatorial structures within Fourier coefficients for image authentication purposes. Cedillo-Hernandez et al. [31] proposed a DFT-based watermarking framework enhanced through particle swarm optimization, improving robustness against common signal-processing attacks. Furthermore, Aksay et al. [32] propose a wavelet-based camera tamper detection approach for video surveillance footage.

## **2.4.2 Spatial Based Watermarking Techniques**

Spatial-based watermarking techniques embed information directly into the host multimedia content by modifying pixel values, sample amplitudes, text attributes, or other structural characteristics without employing frequency transformations. Due to their simplicity, low computational complexity, and high embedding capacity, spatial-domain approaches have been extensively investigated for image, audio, video, and document watermarking applications. In contrast to frequency-domain methods, which require mathematical transformations prior to embedding, spatial techniques operate directly on the original data, making them particularly suitable for real-time systems and computationally constrained environments.

Among the most widely adopted spatial watermarking methodologies are Least Significant Bit (LSB) based techniques. These approaches embed watermark information by modifying the least significant bits of image pixels or audio samples, resulting in minimal perceptual distortion. Kabir et al. [33] introduced an efficient LSB-based watermarking methodology designed to improve both embedding efficiency and ro-

bustness. For video tamper detection, Hasso and Taha [34] propose a low-cost algorithm for video tamper detection. Saglam and Temizel [35] further propose a real-time adaptive algorithm for camera tamper detection in surveillance footage, while Arab et al. [36] propose a robust video watermarking technique specifically designed for tamper detection in surveillance systems.

Block-based spatial watermarking techniques represent another important category of spatial-domain information hiding approaches. Rather than embedding watermark information uniformly throughout the multimedia object, these methods partition the host content into smaller regions and perform localized watermark insertion. Such approaches often provide improved tamper localization capabilities and are particularly attractive for authentication and integrity-preservation applications. By associating watermark information with specific image blocks, unauthorized modifications can frequently be detected and localized with higher accuracy than conventional global watermarking schemes.

One of the principal advantages of spatial watermarking techniques is their computational efficiency. Since embedding and extraction procedures do not require complex mathematical transformations, these approaches generally exhibit lower processing requirements than frequency-domain methods. Furthermore, spatial-domain techniques often provide higher embedding capacities because watermark information can be inserted directly into the host medium. These characteristics make spatial watermarking particularly attractive for large-scale multimedia systems and applications requiring fast watermark generation and verification.

### **2.4.3 ML-based Watermarking Techniques**

The rapid advancement of machine learning and deep learning technologies has significantly influenced the field of digital watermarking. Unlike conventional watermarking approaches that rely on manually designed embedding and extraction rules, machine learning-based techniques utilize data-driven models to automatically learn robust watermark representations and embedding strategies. These approaches aim to improve imperceptibility, robustness, and adaptability by exploiting the feature learning capabilities of neural networks. As a result, machine learning has emerged as a promising paradigm for addressing several limitations associated with traditional watermarking methodologies.

Early learning-based watermarking systems primarily employed convolutional neural networks (CNNs) to improve watermark detection and extraction. By learning discriminative representations from large datasets, CNN-based approaches can adapt to complex distortions and signal-processing operations that often challenge conventional watermarking techniques. Luo et al. [37] proposed a deep-learning-based watermarking framework capable of improving watermark extraction reliability under various distortions. Similarly, Chacko et al. [38] demonstrated that deep neural networks can effectively learn watermark embedding and extraction mappings while maintaining high visual quality.

Recent research has increasingly focused on end-to-end watermarking architectures in which embedding and extraction networks are jointly optimized during training. Such systems typically employ encoder-decoder structures that automatically learn optimal watermark representations and embedding locations. Hayes and Danezis [39] introduced an adversarial learning framework for steganographic image generation, demonstrating the effectiveness of neural networks in hiding information while minimizing perceptual distortion. Similarly, Zhu et al. [40] proposed the HIDDEN framework, which employs deep neural networks to embed and recover information from images while preserving visual fidelity. These approaches illustrated the potential of deep learning to automate watermark design and improve resistance to various distortions.

Another significant research direction involves self-supervised and representation-learning-based watermarking techniques. Fernandez et al. [41] proposed a watermarking framework operating within latent feature spaces generated by self-supervised learning models. Rather than embedding information directly into image pixels, the watermark is inserted within learned feature representations, allowing improved robustness while maintaining high perceptual quality. Such approaches highlight the increasing integration of modern representation learning methodologies into multimedia protection systems.

Generative Adversarial Networks (GANs) have also attracted considerable attention in watermarking research. By employing competing generator and discriminator networks, GAN-based systems can learn watermark embeddings that are both imperceptible and difficult to remove. Several studies have demonstrated that adversarial training can significantly improve watermark robustness against image manipulations and signal-processing attacks. Furthermore, GAN-based approaches can dynamically



adapt embedding strategies based on the characteristics of the host content, providing greater flexibility than traditional fixed-rule watermarking techniques.

Recent machine learning research has additionally explored watermarking for emerging multimedia applications. Sinhal et al. [42] investigated machine learning approaches for secure multimedia protection, while Asmitha et al. [43] examined privacy-preserving watermarking strategies based on intelligent learning mechanisms. More recently, Sood et al. [44], Zhou et al. [45], and Zhang et al. [46] proposed advanced deep-learning-based watermarking frameworks capable of improving watermark robustness, visual quality, and security against sophisticated manipulation attacks.

#### **2.4.4 Hybrid Watermarking Techniques**

More recently, hybrid transform-domain methodologies have emerged as a promising direction in frequency-domain watermarking research. Rather than relying on a single transform, these approaches combine multiple frequency representations to exploit their complementary advantages. Varghese et al. [47] proposed a watermarking framework integrating DCT, DFT, and SVD transformations to improve both robustness and imperceptibility. Similarly, Yuan et al. [48] introduced a blind watermarking scheme that combines multiple transform-domain representations to achieve enhanced extraction reliability. Kumar et al. [49] further demonstrated that hybrid frequency-domain architectures can significantly outperform conventional single-transform approaches under various attack scenarios.

Several researchers have explored the integration of optimization algorithms within hybrid watermarking frameworks. Such approaches employ optimization techniques to identify suitable embedding locations and determine optimal embedding strengths. By dynamically adjusting watermark parameters according to the characteristics of the host content, these methods aim to maximize robustness while minimizing perceptual distortion. The incorporation of optimization procedures has been shown to improve watermark recoverability under a wide range of signal-processing attacks.

Another important category of hybrid watermarking techniques combines spatial-domain and frequency-domain representations. Rather than embedding watermark information exclusively within one domain, these methodologies distribute watermark components across multiple representations of the host content. Such strategies can

improve resilience against attacks targeting a specific domain while simultaneously increasing the difficulty of unauthorized watermark removal. Latreche et al. [50] proposed a secure watermarking framework that combines multiple embedding mechanisms to strengthen content protection and authenticity verification. Similarly, Liu et al. [51] demonstrated that hybrid embedding strategies can improve robustness while preserving acceptable visual quality.

Recent advances have also led to the development of machine learning assisted hybrid watermarking systems. These approaches integrate traditional transform-domain watermarking with intelligent optimization or deep learning models capable of adapting embedding parameters according to the characteristics of the host media. By combining deterministic signal-processing operations with data-driven learning mechanisms, such frameworks seek to leverage the strengths of both conventional and modern watermarking paradigms. Chaudhary et al. [52] and Awasthi et al. [53] demonstrated that hybrid architectures incorporating learning-based components can improve robustness and extraction accuracy under complex attack scenarios.

# CHAPTER 3

## COMPONENTS AND FUNCTIONAL CHARACTERISTICS

- 
- 3.1 Digital Objects Characteristics
  - 3.2 Digital Objects Segmentation Rules
  - 3.3 Frequency Domain Watermarking
  - 3.4 Spatial Domain Watermarking
  - 3.5 Encoding and Hashing Techniques
- 

### 3.1 Digital Objects Characteristics

Digital media such as images, audio, video, and PDF documents possess different structural, perceptual, and storage properties that directly affect the embedding, extraction, robustness, and imperceptibility of hidden information. Characteristics including color representation, sampling rate, frame organization, compression schemes, codecs, and file structure play a significant role in determining how information can be securely embedded while preserving the quality and functionality of the host object. The following subsections briefly describe the basic characteristics and structural properties of the most common digital media categories utilized in information hiding applications.

### 3.1.1 Image Characteristics

Images can initially be categorized based on their color space into RGB, Grayscale, and Binary images. RGB images consist of three separate image components corresponding to the Red, Green, and Blue color channels, which are combined to produce the final colored image. More specifically, each pixel in an RGB image is represented by a triplet of intensity values  $(R, G, B)$ , where each component ranges from 0 to 255. Typical cases of RGB images are generic images, artistic images and forensic images. In the case of grayscale images, the intensity values across the three color channels are equal for each pixel, resulting in image colors limited to different shades of gray. Grayscale images can be found in medical applications where most of the medical image modalities are represented as grayscale images. Binary images, on the other hand contain a single channel, where each pixel takes a value of 0 or 255, representing black and white respectively. QR codes are an example of such images where each pixel of the QR is a binary value (0 or 255). Another important characteristic of digital images is their file format. Among the most commonly utilized formats in watermarking applications are JPEG and PNG, which represent lossy and lossless compression formats respectively.

### 3.1.2 Audio Characteristics

Audio signals can be distinguished according to the number of channels they contain. Mono audio utilizes a single channel, resulting in identical sound reproduction across all output devices, whereas stereo audio employs two distinct channels corresponding to the left and right speakers, allowing spatial representation of sound. Audio information is digitally represented as sequences of sampled signal amplitudes, where the Sampling Rate (SR) directly affects the perceptual quality of the reconstructed audio signal. In addition to channel configuration, audio signals are characterized by their storage and compression format. Among the most commonly utilized audio formats in watermarking systems are MP3 and WAV. MP3 employs lossy compression techniques in order to significantly reduce storage requirements by discarding perceptually less important information, while WAV preserves the original audio content without compression-induced information loss, thus representing a lossless audio format.

### 3.1.3 Video Characteristics

A video object consists of a sequence of consecutive image frames displayed at a specific rate, commonly measured in Frames Per Second (FPS). When the frame rate is sufficiently high, the displayed frames create the perception of continuous motion, making FPS a fundamental characteristic affecting the smoothness and quality of the video. In addition to the visual component, digital videos commonly include synchronized audio streams, which constitute another essential part of the overall structure. Audio data are similarly organized into temporal segments or frames corresponding to specific portions of the audio signal. Depending on the number of audio channels, sound reproduction may be mono, where the same signal is reproduced across all output devices, or stereo, where distinct signals are delivered to the left and right channels, enabling spatial sound perception.

Both video and audio streams are commonly compressed using codecs, which determine the efficiency, storage requirements, transmission capability, and perceptual quality of the video content. Popular video codecs include H.264, H.265, and AV1, while common audio codecs include MP3, AAC, FLAC, and WAV. Depending on the employed codec, compression may be either lossy or lossless, directly affecting the fidelity of the video. Furthermore, video data are stored within container formats, which define how different encoded streams, including frames, audio, subtitles, and metadata, are organized, stored, and synchronized. Common container formats include MP4, AVI, and MKV, each offering different levels of compatibility and compression support.

### 3.1.4 PDF Characteristics

A PDF file follows a structured architecture composed of four main parts: the Header, Body, Cross-Reference Table (xref), and Trailer. The Header identifies the file as a PDF and specifies its version. The Body contains the actual document content, including text, images, fonts, and other objects. Since PDF version 1.5, object streams have also been introduced to reduce file size by grouping multiple objects into compressed streams. The Cross-Reference Table acts as an index that stores the byte offsets of all objects within the file, enabling fast navigation and efficient content retrieval. Finally, the Trailer contains essential metadata, including references to the cross-reference table, document identifiers, and integrity-related information required for correct file inter-

pretation. An important characteristic of the PDF format is its incremental update capability. Instead of rewriting the entire file after each modification, new changes can be appended to the end of the document along with an updated cross-reference table and trailer. This mechanism improves efficiency, preserves previous document states, and facilitates version tracking.

## 3.2 Digital Objects Segmentation Rules

In order to embed the authenticity and integrity information, the digital objects are initially segmented into smaller structural regions according to their type and underlying data representation. This segmentation process enables the proposed watermarking systems to localize the embedding procedure, distribute the watermark information across the digital object, and improve both robustness and tamper localization capabilities. Depending on the type of digital object, the segmentation procedure may involve spatial partitioning of images and video frames into blocks, segmentation of document objects into text regions and character groups, or temporal decomposition of video objects into consecutive frames and synchronized audio streams. The following subsections describe the segmentation rules and partitioning strategies employed for each category of digital object utilized in this research study.

### 3.2.1 Spatial Segmentation

**Image and Frame Authenticity Segmentation:** Initially, the input image/video frame is partitioned into a set of *blocks* based on a predefined *plane*. Let  $I$  be an input image or frame of size  $M \times N$ , which is divided into non-overlapping regions that serve as carriers of the embedded information. For the embedding of authenticity information, each *block* contains a *grid* composed of smaller sub-regions, referred to as *cells*, which correspond to localized fragments of the *block*. Each *cell* consists of a group of *pixels* and serves as the fundamental unit for authenticity information embedding. The proposed authenticity preservation models follow a hierarchical structure defined as follows:

$$Pixel \ll Cell \ll Grid \ll Block \ll Plane \quad (3.1)$$

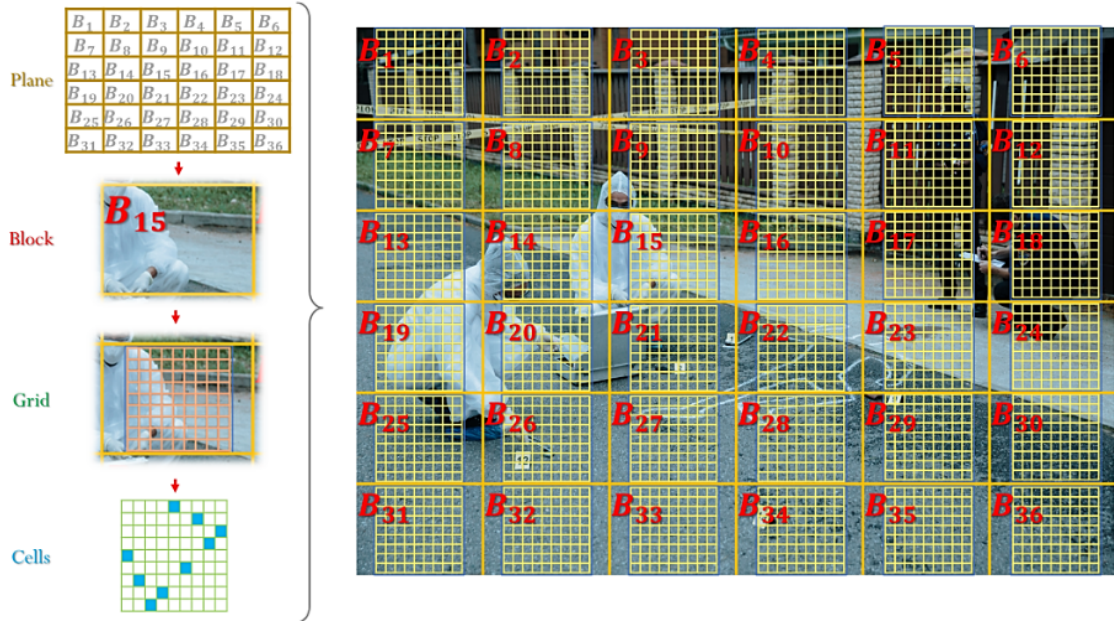


Figure 3.1: Partition of a forensic image into authenticity compartments.

The partition components utilized for the authenticity validation of image or frame, let  $I$  of dimensions  $M \times N$  are listed below:

- **Authenticity-Plane.** An *authenticity-plane* defines particular areas of the image. Specifically, the *authenticity-plane* divides the whole area of the digital image into a number of equal-sized and non-overlapping regions which contain the watermark. In order to calculate the size of the  $d \times d$  plane we select a common divisor of  $(M, N)$  let  $d$ , defining  $d^2$  regions, the so-called *authenticity-blocks*.
- **Authenticity-Blocks.** An *authenticity-block* is the region defined by the division of the *authenticity-plane* by  $d$ , of dimensions  $m \times n$  pixels, such that  $m = \frac{M}{d}$  and  $n = \frac{N}{d}$ . A *authenticity-block* contains the embedded information (i.e., the watermark) that will serve as the authenticity validation component.
- **Grid.** A *grid* is defined inside the area of a *authenticity-block* as a rectangular area in the center of the block of dimensions  $m' \times m'$  pixels, such that  $m' = \min\{m, n\}$  defining  $c^2$  regions, the so-called *cells*.
- **Cells.** A *cell* is a fundamental component of the proposed image partition approach, defining a rectangular sub-area of a *grid* that stores the hidden information required for watermarking. A *cell*, has dimensions  $\frac{m'}{c} \times \frac{m'}{c}$  pixels, where  $c$  is an odd number.

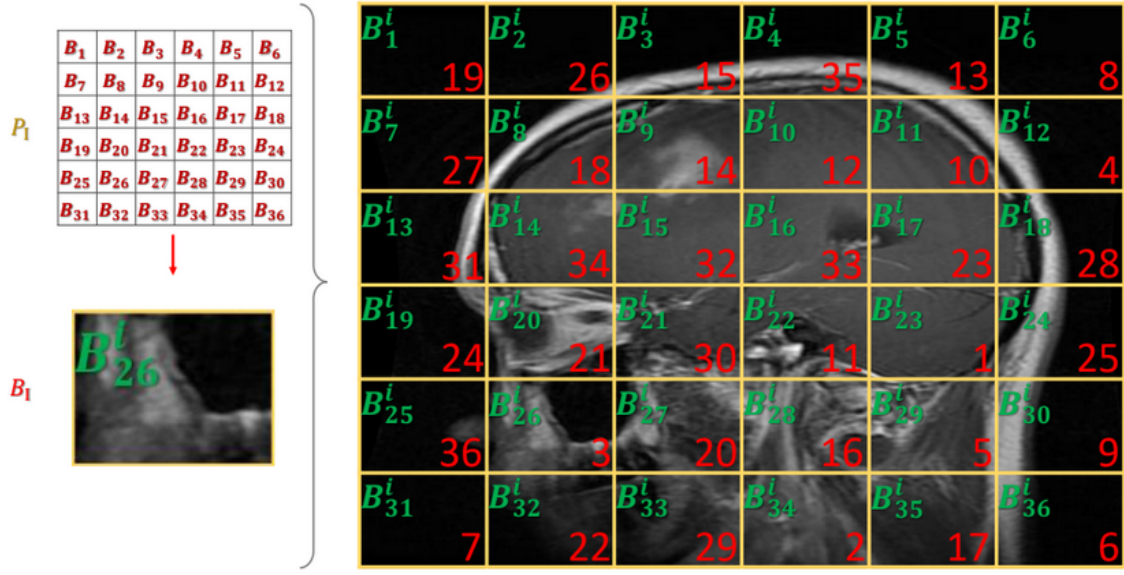


Figure 3.2: Partition of a medical image into integrity compartments.

In Figure 3.1 we depict the segmentation of image/frame to the related compartments in order to embed the authenticity information.

**Image Integrity Segmentation:** For the embedding of integrity information, the digital image is partitioned into discrete structural regions referred to as *integrity-blocks*. It should be noted that, in the context of the proposed work, the integrity-preservation mechanism is applied exclusively to digital images and not to individual video frames. Given an image  $I$  of dimensions  $M \times N$ , the image is divided into spatial regions, where each block contains a group of neighboring pixels corresponding to a specific area of the image. More specifically, the partitioning scheme utilized for the embedding of integrity information is defined as follows:

- **Integrity-Plane.** An *integrity-plane* defines specific areas of the medical image by dividing it into a number of equal-sized, non-overlapping regions that contain the integrity information. Given an integer  $k$ , the *integrity-plane* is defined as a  $k \times k$  partition, resulting in  $k^2$  non-overlapping *blocks* of equal size, referred to as *integrity-blocks*.
- **Integrity-Blocks.** An *integrity-block* is a region defined by the partitioning of the *integrity-plane*, with dimensions  $m \times n$ , where  $m = \frac{M}{k}$  and  $n = \frac{N}{k}$ . Each *integrity-block* contains  $m \times n$  pixels, whose values across the RGB channels are utilized



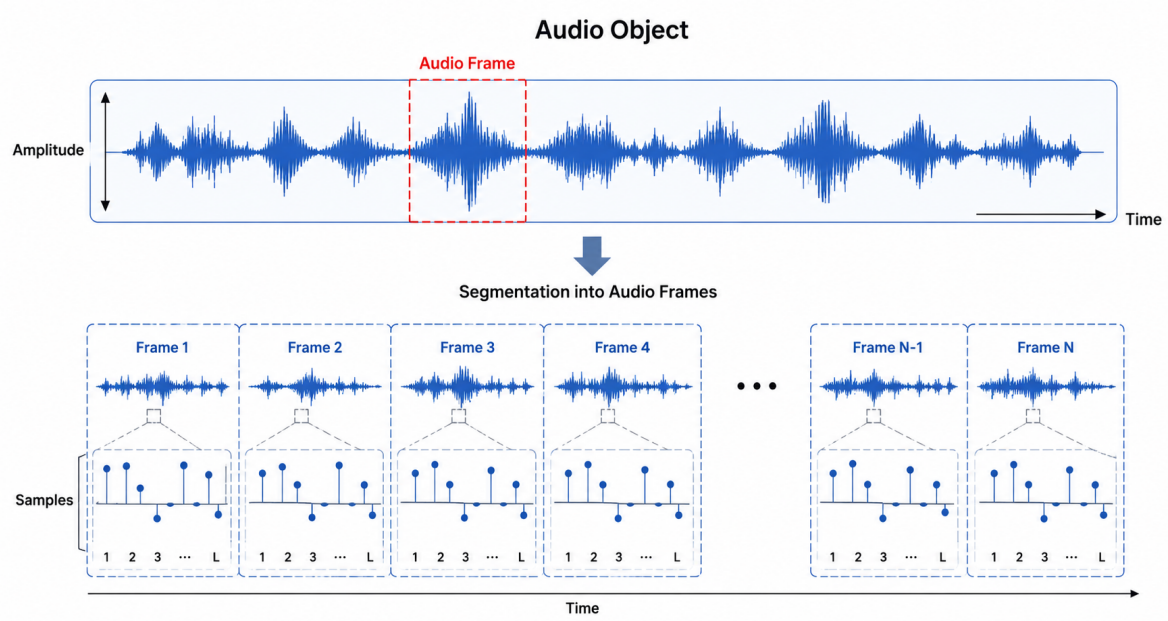


Figure 3.3: Partition of an audio signal into the proposed compartments.

for integrity verification (in particular, those of the red and green channel) to compute and store hash values.

In Figure 3.2 we depict the segmentation of a medical image to the related compartments in order to embed the integrity information.

**PDF Segmentation:** In the case of PDF documents, the segmentation procedure is performed according to the textual structure of each document page. Initially, the textual body of the PDF page is extracted from the document object. Subsequently, the extracted content is partitioned into *Text Fragments*, where each fragment corresponds to a discrete textual element of the document. The collection of all extracted fragments forms the *Text Fragment Collection* associated with the corresponding PDF page. Finally, each *Text Fragment* is decomposed into individual characters, which constitute the fundamental embedding units utilized by the proposed watermarking framework. The hierarchical segmentation structure of the proposed PDF watermarking model is therefore defined as follows:

- **PDF:** Represents the complete PDF document object.
- **Page:** Represents an individual page of the PDF document.
- **Text Fragment Collection:** Represents the collection of all extracted text fragments objects associated with a specific PDF page.

- **Text Fragment:** Represents a discrete textual element extracted from the PDF structure.
- **Character:** Represents the individual character corresponding to the fundamental embedding unit utilized by the proposed PDF watermarking framework.

### 3.2.2 Temporal Segmentation

**Video Segmentation:** Retrieving image frames refers to the process of accessing individual frames from a digital video sequence for analysis and processing purposes. In the proposed framework, this procedure plays a fundamental role during both the embedding and extraction phases. During embedding, the video is decomposed into frames, enabling the generation and insertion of the watermark information into the corresponding frame regions before reconstructing the final watermarked video. Conversely, during extraction and verification, the same frame decomposition process is performed in order to retrieve the embedded watermark information, allowing the authenticity and integrity of the video content to be assessed. The process involves loading the video sequence and accessing specific frames according to temporal positions determined through FPS of the video.

Similarly, audio frame retrieval involves extracting the audio stream associated with the digital video object for independent analysis and processing. In the proposed work, audio frames are utilized for the storage of integrity-related information associated with the watermarked video. This process requires isolating the audio stream from the video object, accessing the raw audio samples, and performing the necessary preprocessing operations independently from the visual component of the video sequence.

**Audio Segmentation:** Audio data, originating either from individual audio files or from audio streams associated with video objects, are decomposed based on their channel arrangement. Let  $A$  represent an audio signal composed of one or multiple channels, where each channel corresponds to an ordered sequence of sampled amplitude values. Depending on the audio representation, the signal may consist of a mono configuration containing a single channel or a multi-channel configuration, such as stereo audio with separate left and right channels. The hierarchical organization of the proposed audio segmentation model is defined as follows:



Figure 3.4: Partition video objects into image frames and corresponding audio sequence.

- **Audio-Sequence.** An *audio-sequence* refers to the complete sequence of audio samples representing the entire audio signal.
- **Audio Channels.** Each audio object may contain one or multiple audio channels. In the case of mono audio, a single *audio-sequence* is considered, whereas in stereo audio two distinct *Audio Sequences* are utilized, each corresponding to one audio channel.
- **Audio-Frame.** An *audio-frame* corresponds to a temporal segment of an *Audio Sequence*, primarily utilized in digital video objects, where each audio frame is associated with a corresponding visual frame.
- **Audio-Samples.** *audio-sequences* and *audio-frames* are composed of *audio-samples*, which represent discrete amplitude values of the audio signal. In the case of mono audio, each sample corresponds to a single amplitude value, whereas in stereo audio each sample consists of a pair of values corresponding to the left and right audio channels respectively.

In Figure 3.4 we depict the partition of video in image frames and audio sequence, while in Figure 3.3 we depict the partition of an audio signal to the proposed audio

structure in order to embed authenticity or integrity information.

### 3.3 Frequency Domain Watermarking

Frequency-domain watermarking refers to the transformation of digital data, such as an image or video frame, from the spatial (or time) domain into the frequency domain using mathematical transformation functions in order to embed the watermark. Among the most commonly utilized frequency transformation techniques are the FFT, DCT, and DWT. In the proposed work, the embedding of authenticity information into images and video frames is performed in the frequency domain using the FFT. More specifically, the watermark is embedded within the high-frequency FFT coefficients. The selection of high-frequency coefficients for watermark embedding is motivated by the following considerations:

1. **Imperceptibility:** High-frequency components represent rapid variations in pixel intensity, such as edges and fine details, to which the human visual system is generally less sensitive. Embedding the watermark within these regions helps ensure that the introduced modifications remain visually imperceptible, thus preserving the perceptual quality of the video frames.
2. **Avoiding Low-Frequency Distortion:** Low-frequency coefficients, including the DC component, contain the most visually significant information, such as overall illumination and smooth intensity variations. Modifications in these regions are more likely to produce visible artifacts. Consequently, embedding within higher-frequency coefficients reduces the risk of noticeable distortion.
3. **Consistency with Frequency-Domain Watermarking Principles:** Existing watermarking studies have demonstrated that high-frequency embedding can achieve a satisfactory balance between imperceptibility and robustness when combined with redundancy and error-tolerant extraction mechanisms.

Given an image or video frame  $f$  of dimensions  $M \times N$ , its two-dimensional Fast Fourier Transform can be expressed as:

$$F(u, v) = \sum_{x=0}^{n-1} \sum_{y=0}^{m-1} f(x, y) e^{-j2\pi(\frac{ux}{n} + \frac{vy}{m})}, \quad (3.2)$$

where,  $F(u, v)$  represents the value at frequency coordinates  $(u, v)$ ,  $f(x, y)$  represents the pixel value at spatial coordinates  $(x, y)$ ,  $n$  represents the number of rows of the image/frame,  $m$  represents the number of columns of the image/frame,  $j$  is the imaginary  $\sqrt{-1}$ ,  $u \in [0, 1, \dots, n-1]$  represents the horizontal frequency index, and  $v \in [0, 1, \dots, m-1]$  represents the vertical frequency index.

Similarly given  $F(u, v)$  we can use the inverse Fourier Transformation to transform data from the frequency domain back to the spatial domain (see, Equation 3.3).

$$F(u, v) = \frac{1}{nm} \sum_{x=0}^{n-1} \sum_{y=0}^{m-1} f(x, y) e^{-j2\pi(\frac{ux}{n} + \frac{vy}{m})}, \quad (3.3)$$

After applying the FFT, the transformed signal is represented by complex frequency coefficients consisting of a real and an imaginary part. These components can be expressed in terms of magnitude and phase information. The magnitude represents the strength or amplitude contribution of each frequency component, while the phase describes the positional information of the corresponding frequencies within the signal. In the proposed work, the embedding of authenticity information is performed by modifying the magnitude of selected high-frequency FFT coefficients, while preserving the corresponding phase information in order to minimize perceptual distortion.

### 3.4 Spatial Domain Watermarking

The LSB substitution is a fundamental concept in digital data representation and is widely utilized in information hiding applications due to its minimal impact on perceptual quality. Modifications at the LSB level introduce only negligible distortion to the host signal, allowing the embedded information to remain visually imperceptible to the human eye. This characteristic makes LSB-based embedding particularly suitable for fragile watermarking applications, where the hidden information must remain unobtrusive while being highly sensitive to even minor modifications of the host content. In the proposed work, LSB embedding is utilized in images, audio signals, and video frames for the insertion of integrity-related information. More specifically, modifications are applied to selected LSBs of the corresponding digital samples or pixel intensity values, enabling information embedding while preserving the perceptual quality of the host media.



Figure 3.5: Applying the forward and the inverse Arnold Transformation on a QR code.

In PDF documents, spatial-domain watermarking techniques operate directly on the visible characteristics of the document without transforming the content into another mathematical domain. In the proposed work, watermark information is embedded through controlled modifications of the RGB color values of selected text characters within the PDF document. More specifically, QR-code elements are mapped to specific document characters, and characters corresponding to selected QR pixels are assigned slightly modified color values, while the remaining characters preserve their original appearance. These modifications are visually imperceptible to the human eye, preserving the readability and perceptual quality of the document. To further enhance security, the Arnold Transformation is utilized in order to spatially rearrange the pixels of the input data (in our proposed model a QR code), producing a scrambled representation that cannot be easily recognized visually. In the context of digital watermarking, the Arnold Transformation operates as an additional security layer during the embedding procedure, since it randomizes the spatial distribution of the watermark information, making unauthorized detection, interpretation, or manipulation significantly more difficult. The mathematical formulation of the forward and inverse Arnold Transformation is presented below:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \mod N \quad (3.4)$$

where  $(x, y)$  denotes the original pixel coordinates,  $(x', y')$  the transformed coordinates, and  $N$  the dimension of the square image.

The inverse Arnold Transformation is defined as follows:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} \mod N \quad (3.5)$$

where  $(x', y')$  denotes the pixel coordinates after the utilizing forward Arnold Trans-

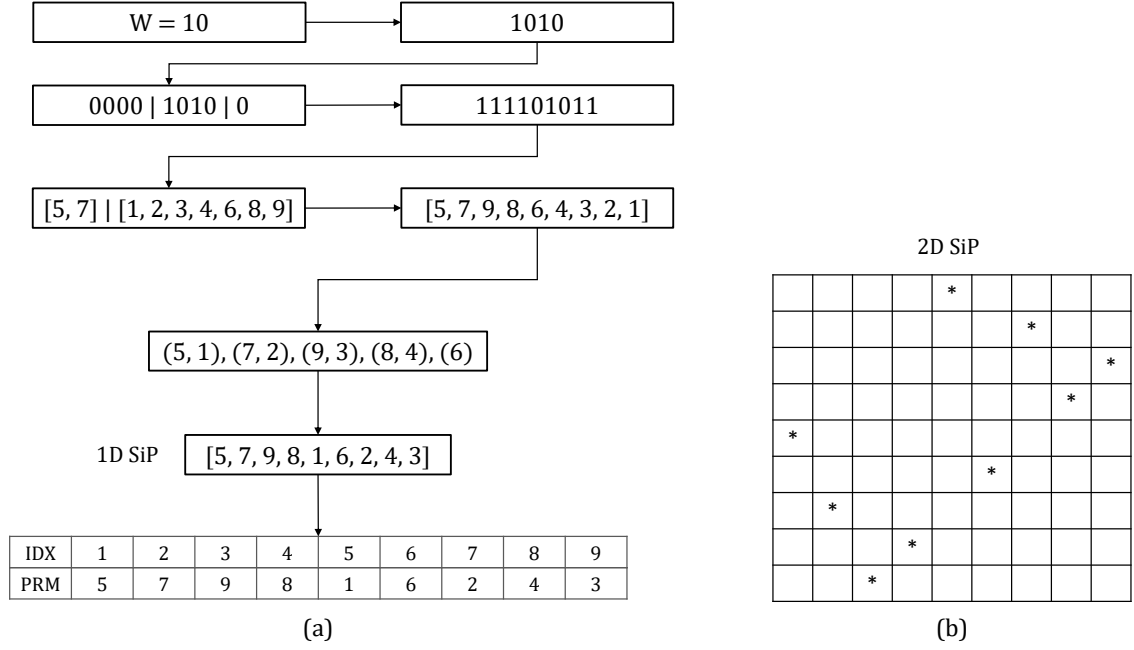


Figure 3.6: Encoding of an integer  $W$  as a SiP. (a) Illustration of the encoding of an integer  $W$  into a Self-inverting Permutation (SiP) and its corresponding mapping. (b) The resulting 2D representation of the SiP, where the  $*$  symbols denote the marked cells carrying the embedded information.

form,  $(x, y)$  the original coordinates, and  $N$  the dimension of the square image. Figure 3.5 depicts the Arnold Transformation (forward and inverse) when it is applied on a QR code.

### 3.5 Encoding and Hashing Techniques

In this section, the fundamental information hiding techniques forming the basis of the proposed watermarking models are presented, covering both robust authenticity watermark embedding and fragile integrity information embedding.

#### 3.5.1 Encoding Integers to Self-Inverting Permutations

The embedding of integer-based watermark information requires its conversion into a structured representation that can be efficiently mapped onto a two-dimensional object like image or video frame. In the proposed methodology, this conversion is achieved through the utilization of SiPs, whose properties support both efficient encoding and

reliable reconstruction during the extraction process.

More specifically, a given integer is initially transformed into a corresponding SiP according to the encoding procedure described in [1]. This representation guarantees that the original integer can be uniquely recovered through the inverse decoding process. In order to embed the generated SiP into the host image/frame, a two-dimensional representation is constructed in the form of a *grid* of dimensions  $k \times k$ , where  $k$  denotes the length of the permutation. The *grid* is further divided into  $k^2$  *cells*, and specific *cells* are selected to host the embedded watermark. More precisely, each row of the *grid* corresponds to an index position of the permutation, while the permutation value at the corresponding index determines the column position of the selected *cell*. Consequently, exactly one *cell* is selected in each row, forming a unique spatial arrangement representing the encoded permutation. An illustrative example of the encoding procedure is presented in Figure 3.6, where an integer  $W$  is transformed into its corresponding SiP and subsequently mapped into 2D representation. The selected *cells* of the *grid* determine the spatial locations where the watermark embedding process is applied. Next we describe in depth the embedding and extracting procedures that form the basis for the proposed models.

### Embedding Procedure

Let an image block  $B_i$  and a SiP of length  $n$ . The block  $B_i$  is partitioned with a *grid* of dimensions  $n \times n$  in  $n^2$  *cells*. The 2D representation of the integer  $w$  we want to embed determines the *Cells* to be marked within the *grid*. For each selected *cell*, the corresponding image region is processed in the frequency domain. Specifically, the Fast Fourier Transform (FFT) is applied, producing the magnitude and phase components, denoted as  $M_{ij}$  and  $P_{ij}$ , respectively. Two annular regions are defined on the magnitude spectrum, referred to as the Red ( $R_{ij}$ ) and Blue ( $B_{ij}$ ) annuli. The average magnitude values within these regions are computed as  $Avg(R_{ij})$  and  $Avg(B_{ij})$ . For each row of the *grid*, the difference

$$D_{ij} = |Avg(B_{ij}) - Avg(R_{ij})|, \quad \text{for } Avg(B_{ij}) \leq Avg(R_{ij}), \quad (3.6)$$

is evaluated, and the maximum value  $Max(D_i)$  across the row is retained. The selected (marked) *cell* is then modified by increasing its magnitude values according to:



$$M'_{ij} = Avg(B_{ij}) + Avg(R_{ij}) + Max(D_i) + c, \quad (3.7)$$

where  $c$  is a user-defined strength parameter controlling the embedding intensity. The modified magnitude component is then recombined with the original phase component, and the inverse FFT (iFFT) is applied to reconstruct the spatial-domain representation of the *cell*

### Extracting Procedure

Given a watermarked block  $Bw_i$ , we use the same *grid* of dimensions  $n \times n$  to partition the block into  $n^2$  *cells*. For each *cell*, the FFT is applied to obtain the magnitude and phase components. Utilizing the same annular regions, the average values  $Avg(R_{ij})$  and  $Avg(B_{ij})$  are computed. For each row of the *grid*, the difference

$$|Avg(B_{ij}) - Avg(R_{ij})| \quad (3.8)$$

is evaluated for all *Cells*, and the *cell* that minimizes this value is selected as the marked position for each row of the *Grid*, to finally form the extracted SiP which will be decoded to an Integer that must match with the one we embedded to the specific block.

### 3.5.2 Integrity Preservation Utilizing Hash Function

One of the most widely used approaches for integrity preservation in digital images is the computation of cryptographic hash values. The hash value is typically a fixed-length hexadecimal string that effectively represents the data given as input to the hash function. A key advantage of hash-based methods is that hash values can be compared significantly faster and more efficiently than the original image data, making them suitable for applications such as database management and image retrieval. In the context of integrity verification, cryptographic hash functions are particularly effective for tamper detection, as even a minimal modification, such as a change in a single pixel, results in a significantly different hash value, thereby enabling effective tamper detection and localization. Below we present the key properties of cryptographic hash functions:

- **Fixed Length:** Regardless of the size of the input data, the produced SHA- $X$  hash always has a fixed length of  $X$  bits.
- **Deterministic:** The same input data will always generate the exact same hash value whenever the hashing process is repeated.
- **Avalanche Effect:** A minimal modification in the input data, even a single character or bit, produces a completely different and unpredictable hash value.
- **Irreversibility:** The original input data cannot be reconstructed from the generated SHA- $X$  hash value, making the hashing process computationally one-way.

Based on the above properties, SHA-256 was selected in the proposed work for tamper detection and localization purposes. More specifically, the deterministic nature of SHA-256 guarantees that identical digital content always produces the same hash value, enabling reliable integrity verification. Furthermore, the avalanche effect ensures that even minimal modifications within the protected digital object produce completely different hash values, allowing efficient localization of tampered regions. The fixed-length representation of SHA-256 ensures stable computational efficiency, since different input data always produce hash values of constant length, resulting in a fixed-size integrity watermark representation. The irreversible nature of SHA-256 prevents malicious users from reconstructing the original content from the embedded hash information. These properties collectively make SHA-256 highly suitable for fragile watermarking applications focusing on integrity preservation, tamper detection, and tamper localization.

## CHAPTER 4

# MODELS: AUTHENTICITY AND INTEGRITY

---

### 4.1 Authenticity: Embedding Robust Information

### 4.2 Integrity: Embedding Fragile Information

---

## 4.1 Authenticity: Embedding Robust Information

In this section, the proposed watermarking models developed for the preservation of authenticity in digital images, audio files, and digital video objects are presented and analyzed. More specifically, the embedding and extraction methodologies of each model are described, along with the corresponding information hiding mechanisms utilized for robust authenticity verification. The proposed approaches are designed with emphasis on robustness, security, and imperceptibility, aiming to preserve the authenticity while maintaining high perceptual quality.

### 4.1.1 Images

#### Image Watermarking Model utilizing Repetitive SiP Embedding

As a starting point of the present dissertation, the watermarking system based on SiPs was studied [1], where a single integer in the form of SiP is embedded into specific regions of the image (see Figure 4.1). However, it was observed that this limited spatial coverage of the digital object may introduce vulnerabilities, since attacks such as cropping regions of the image that do not contain embedded watermark information cannot be reliably verified with respect to their authenticity.

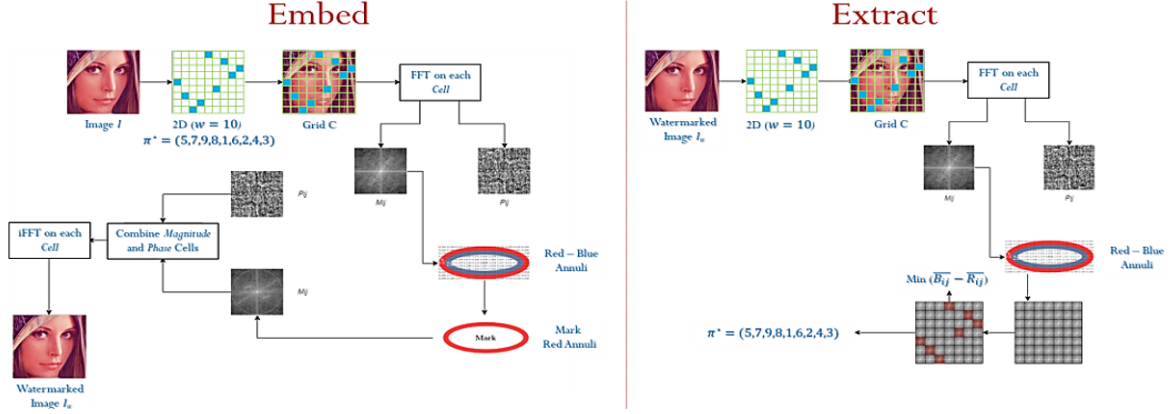


Figure 4.1: Embedding and Extracting procedures utilizing Self-Inverting permutations forming the basis of our proposed image watermarking models.

In order to address these limitations, the embedding strategy was modified through a repetitive and wider distribution of the SiP across multiple image blocks. More specifically, instead of the conventional approach where a single integer derived from a SiP is embedded across the entire image, with each permutation element corresponding to a specific image block, the proposed methodology performs repetitive embedding of the same SiP into multiple image blocks. In this manner, the watermark information is replicated throughout multiple spatial regions of the image, enabling successful reconstruction of the embedded watermark even after severe cropping and compression attacks by utilizing the remaining unaffected regions of the image.

Algorithm 4.1 presents the repetitive watermark embedding procedure of the proposed approach. Initially, the 2DM representation of SiP encoding the integer  $w$  is constructed and utilized in order to determine the spatial locations of the *authenticity-blocks*  $B_{1,\pi^*(1)}, B_{2,\pi^*(2)}, \dots, B_{\ell(\pi^*),\pi^*(\ell(\pi^*))}$  within the image  $I$ . More specifically, the marked positions of the 2DM representation define the image blocks that will be used in the watermark embedding procedure, as illustrated in Figure 4.2. Subsequently, the same 2DM representation is utilized within each selected block in order to identify the corresponding *cells*  $C'_{1,\pi^*(1)}, C'_{2,\pi^*(2)}, \dots, C'_{\ell(\pi^*),\pi^*(\ell(\pi^*))}$  where the watermark information will be embedded. The FFT transformation is then applied to each selected *cell* in order to compute the corresponding amplitude matrices. The

---

**Algorithm 4.1** Repetitive Embedding Procedure

---

```
1: Input: Image  $I$ , Integer  $w$ 
2: Output: Watermarked image  $I_w$ 
3: Construct the 2DM representation of  $Sip(w)$ 
4: for each marked cell in the 2DM representation do
5:   Select the corresponding image blocks
6:   for each selected image block  $B_{i,\pi^*(i)}$  do
7:     Construct the corresponding 2DM representation
8:     for each marked cell in the 2DM representation do
9:       Select the corresponding sub-cells  $C'_{i,j}$ 
10:      for each selected cell  $C'_{i,j}$  do
11:        Compute the FFT transformation
12:        Extract the Magnitude matrix  $M_{ij}$  and Phase matrix  $P_{ij}$ 
13:        Place the Red-Blue Annuli on  $M_{ij}$ 
14:        if  $C'_{i,j}$  corresponds to a marked cell then
15:          Mark the Red Annuli
16:        end if
17:        Combine  $M_{ij}$  with  $P_{ij}$ 
18:        Compute the inverse FFT (iFFT)
19:      end for
20:    end for
21:  end for
22: end for
23: Construct the final watermarked image  $I_w$ 
24: Return the final watermarked image  $I_w$ 
```

---

Red and Blue Annuli are placed on these matrices, and the *cells* selected by the 2DM representation of the integer  $w$  have their Red Annuli marked. This procedure is repeated for all selected *cells* within each *authenticity-block*. Finally, the watermarked image is reconstructed by applying the inverse Fast Fourier Transformation (iFFT) to the modified *cells*.

In Algorithm 4.2, the repetitive watermark extraction procedure of the proposed approach is presented step-by-step. It is important to note that the proposed repetitive embedding strategy ensures that the successful extraction of at least one embedded

---

**Algorithm 4.2** Repetitive Extraction Procedure

---

```
1: Input: Watermarked Image  $I_w$ 
2: Output: List of Extracted integer watermarks  $w$ 
3: Initialize empty list  $L_w$ 
4: Construct the corresponding 2DM representation
5: for each marked cell in the 2DM representation do
6:   Select the corresponding image blocks from  $I_w$ 
7:   for each selected image block  $B_{i,\pi^*(i)}$  do
8:     Construct the corresponding 2DM representation
9:     for each marked cell in the 2DM representation do
10:      Select the corresponding sub-cells  $C'_{i,j}$ 
11:      for  $i = 1$  to  $\ell(\pi^*)$  do
12:        Compute the FFT transformation
13:        Extract the Magnitude matrix  $M_{ij}$  and Phase matrix  $P_{ij}$ 
14:        Place the Red-Blue Annuli on  $M_{ij}$ 
15:        Select the cell satisfying:
16:          
$$Avg_{B_{ij}} - Avg_{A_{ij}} = \min \{ Avg_{B_{ij}} - Avg_{A_{ij}} \}_{j=1}^{\ell(\pi^*)}$$

17:        Construct the corresponding SiP
18:        Decode the extracted SiP to get the Integer  $w$ 
19:        Store  $w$  to  $L_w$ 
20:      end for
21:    end for
22:  end for
23: Return list  $L_w$ 
```

---

watermark instance is sufficient to verify the authenticity of the image  $I_w$ . Initially, the 2DM representation of the SiP encoding the integer  $w$  is utilized in order to determine the *authenticity-blocks* within the watermarked image  $I_w$ , as illustrated in Figure 4.3. More specifically, the marked positions of the 2DM representation define the image regions that potentially contain the embedded watermark information. Consequently, the corresponding cells  $B_{1,\pi^*(1)}, B_{2,\pi^*(2)}, \dots, B_{\ell(\pi^*),\pi^*(\ell(\pi^*))}$  are identified and selected for the extraction procedure. Subsequently, the same repetitive extraction process is applied within each selected region by utilizing once again the 2DM

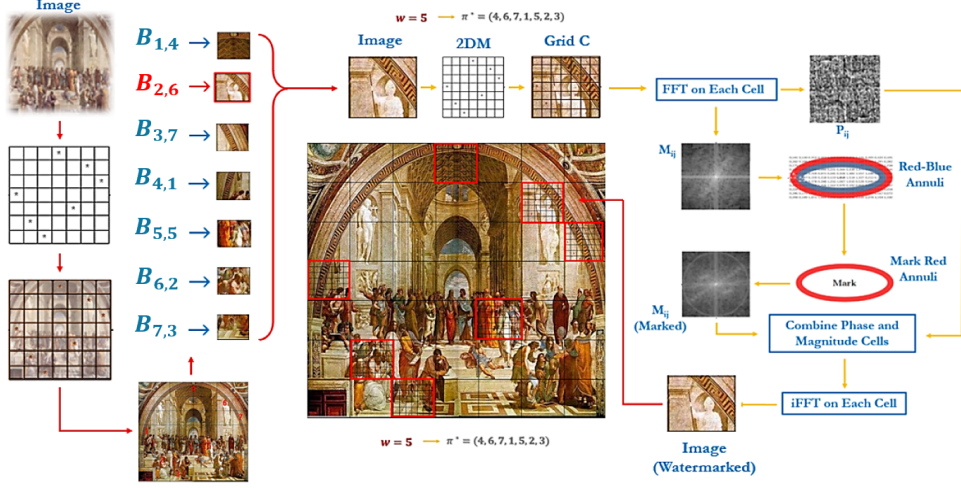


Figure 4.2: Repetitive Embed procedure utilizing Self-Inverting Permutations.

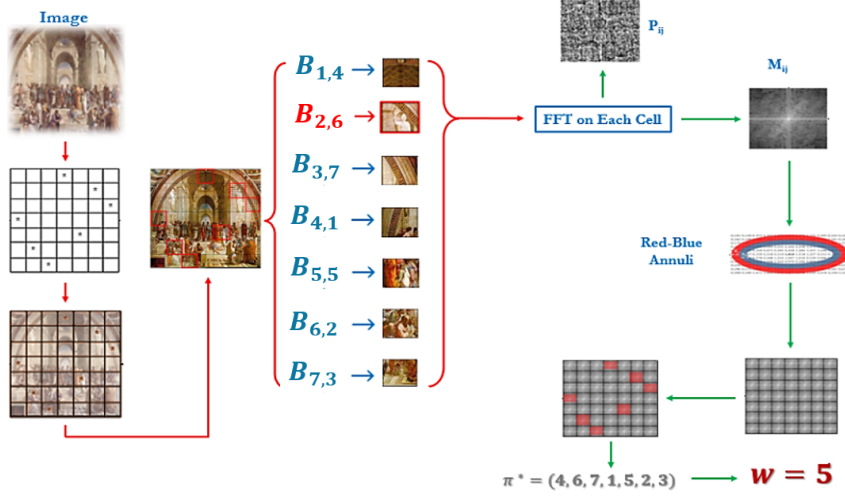


Figure 4.3: Repetitive Extract procedure utilizing Self-Inverting Permutations.

representation of the embedded SiP. More specifically, the corresponding sub-cells  $C'_{1,\pi^*(1)}, C'_{2,\pi^*(2)}, \dots, C'_{\ell(\pi^*),\pi^*(\ell(\pi^*))}$  are identified within the boundaries of each selected block  $B_{i,j}$ . The FFT transformation is then applied to each selected sub-cell in order to compute the corresponding amplitude matrices utilized during the extraction process. By analyzing the marked frequency regions of the amplitude matrices, the embedded SiP is reconstructed. The same procedure is repeated for all selected cells within the identified *authenticity-blocks*. Finally, the extracted Self-inverting Permutation of each block is decoded in order to reconstruct the embedded integer watermark  $w$ .

### Code-based Multiple SiP Image Watermarking Model

---

**Algorithm 4.3** Code-based Multiple Embedding Procedure

---

```
1: Input: Image  $I$ , integers  $d, c$ , range  $R$ 
2: Output: Watermarked image  $I_w$ 
3: Split the image into  $d^2$  image blocks
4: for each image block  $B_i$ , where  $1 \leq i \leq d^2$  do
5:   Randomly select an integer:  $S_c[i] \leftarrow \text{rand}(\min(R), \max(R))$ 
6:   Assign:  $w \leftarrow S_c[i]$ 
7:   Construct the 2DM representation  $[SiP(w)]_2$ 
8:   for each cell  $c_{ij}$  in  $[SiP(w)]_2$  do
9:     if  $c_{ij}$  corresponds to a marked cell then
10:      Select the corresponding cell from the image block  $B_i$ 
11:      Compute the FFT transformation
12:      Extract the Magnitude matrix  $M_{ij}$  and Phase matrix  $P_{ij}$ 
13:      Place the Red-Blue Annuli on  $M_{ij}$ 
14:      Mark the Red Annuli of  $M_{ij}$ 
15:      Combine  $M_{ij}$  with  $P_{ij}$ 
16:      Compute the inverse FFT (iFFT)
17:     end if
18:   end for
19: end for
20: Return watermarked image  $I_w$ 
```

---

Despite the improvement achieved from the previous repetitive approach, the problem of incomplete spatial coverage of the image still remained. For this reason, the proposed approach was further extended through the utilization of multiple SiPs, which are systematically embedded across all image blocks, achieving full spatial coverage of the digital image.

More specifically for a given image  $I$ , we select an *authenticity-plane* of dimensions  $d \times d$  that results to  $d^2$  *authenticity-blocks*. We define a sequence  $S$  of length  $|S_c| = d^2$  integers that later will be represented by Self-Inverting Permutations of size  $P$ , such that  $P$  is an odd integer. For each *authenticity-block* resulting from the partitioning of  $I$  using the *authenticity-plane*, we define a corresponding  $(m' \times m')$  grid and mark particular *cells* of it (note that  $|cells| = P$ ) in order to embed an integer. Each integer of the code-sequence is embedded in the corresponding block as follows:



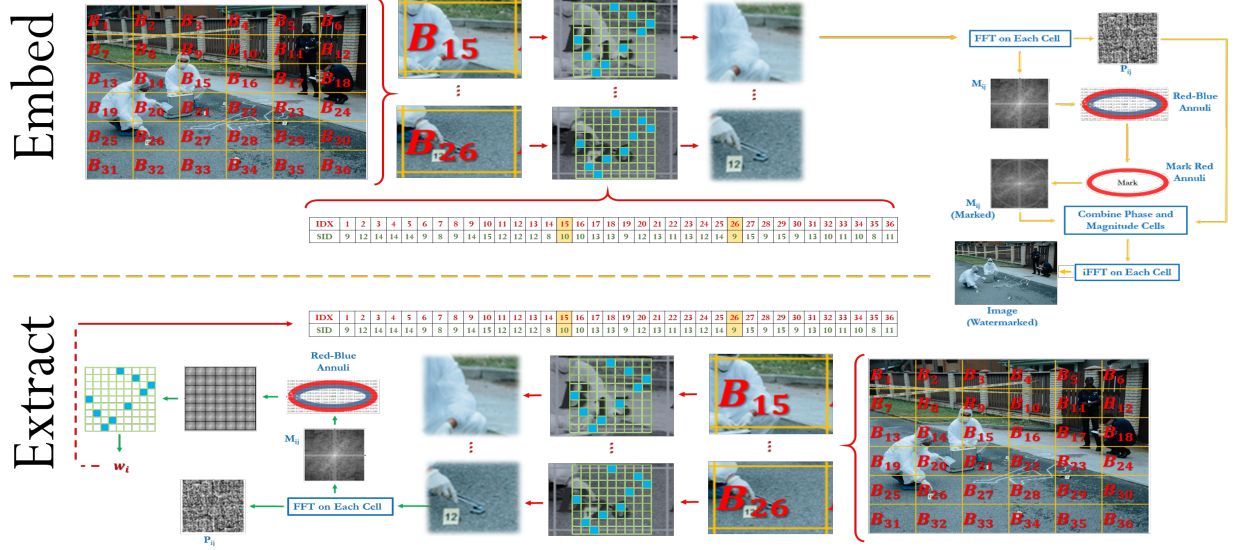


Figure 4.4: Embed and Extract procedure of the image code-based multiple watermarking scheme.

$$G(B_i) \leftarrow [SiP(S_c[i])]_2, \quad (4.1)$$

where,  $G(B_i)$  corresponds to the particular *grid* located inside the boundaries of the  $i^{th}$  *authenticity-block*, i.e.,  $B_i$ ,  $SiP(S_c[i])$  corresponds to the Self-Inverting Permutation of the  $i^{th}$  integer of the sequence  $S$ , i.e.,  $S_c[i]$ , and  $[\cdot]_2$  correspond to the 2D representation of  $S_c[i]$ . Next, for each cell, let  $c_{ij}$ , we follow the process defined in 3.5.1 in order to embed each integer of  $S_c$  to the corresponding blocks. In Algorithm 4.3 we present the steps of the embedding algorithm.

For the extraction procedure, we utilize the watermarked image  $I_w$ , the integers  $d$  and  $c$  that correspond to the *authenticity-plane* and *grid* size, respectively, and the code-sequence  $S_c$  that contains the embedded integers in order to finally compute the extraction rate  $R_e$  to determine the authenticity of the forensic image. Initially, we partition the watermarked image using a *authenticity-plane* defining  $d^2$  *authenticity-blocks*, where from the *grid* placed inside each *authenticity-block* we will extract the embedded integers denoted by the marked *cells*. For each *authenticity-block* the extraction procedure described in Section 3.5.1 is utilized in order to reconstruct the corresponding SiP and consequently recover the embedded integer  $w$ , which is then appended to the extracted code-sequence  $S_c^e$ . Finally, once the extracted code-sequence  $S_c^e$  has been completed we compare the sequence of integers in  $S_c^e$  with the corresponding integers into  $S_c$  on the same positions to compute the extraction rate  $R_e$ . The authenticity of

---

**Algorithm 4.4** Code-based Multiple Extraction Procedure

---

```
1: Input: Watermarked image  $I_r$ , integers  $d, c$ , code-sequence  $S_c$ 
2: Output: Extraction rate  $R_e$ , extracted sequence  $S_c^e$ 
3: Define and empty list  $S_c^e$ 
4: Split the image into  $d^2$  image blocks
5: for each image block  $B_i$ , where  $1 \leq i \leq d^2$  do
6:   for each line  $\ell$  in  $G(B_i)$  do
7:     Initialize:  $min \leftarrow 0, \quad pos \leftarrow 0$ 
8:     for each cell  $c_{ij}$  in line  $\ell$  do
9:       Compute the FFT transformation
10:      Extract the Magnitude matrix  $M_{ij}$  and Phase matrix  $P_{ij}$ 
11:      Place the Red-Blue Annuli on  $M_{ij}$ 
12:      if  $(Avg_{B_{c_{ij}}} - Avg_{R_{c_{ij}}}) < min$  then
13:         $min \leftarrow (Avg_{B_{c_{ij}}} - Avg_{R_{c_{ij}}})$ 
14:         $pos \leftarrow j$ 
15:      end if
16:    end for
17:    Mark position  $(\ell, pos)$  in  $[SiP(w_i)]_2$ 
18:  end for
19:   $S_c^e[i] \leftarrow [SiP(w_i)]_2$ 
20:  if  $S_c^e[i] = S_c[i]$  then
21:     $counter \leftarrow counter + 1$ 
22:  end if
23: end for
24:  $R_e \leftarrow (counter/|S_c|) \times 100$ 
25: Return  $R_e, S_c^e$ 
```

---

the examined image is determined based on the extracted code-sequence  $S_c^e$  and the corresponding extraction rate  $R_e$ , where  $0 \leq R_e \leq 100$ . In Algorithm 4.4 we present the steps of the extraction algorithm.

The proposed multiple watermarking scheme provides a strong level of security, as an attacker would need to infer several unknown parameters and perform extensive trial-and-error attempts in order to recover the embedded information. Specifically, the attacker must determine both the size  $d \times d$  of the *authenticity-plane* and the size

$P$  of the 2D representation associated with the Self-inverting Permutation. More formally, the proposed scheme constructs a code-sequence  $S_c$  of length  $n$ , where each element is an integer in the range  $R = [p, q]$  such that  $|R| = 2^i$ . While the number of distinct permutations of the elements in  $R$  is  $(2^i)!$ , allowing repetitions in  $S_c$  results in a total of  $(2^i)^n$  possible code-sequences. This exponentially large search space significantly increases the difficulty of unauthorized reconstruction of the embedded information. More specifically we construct a sequence of integers in the range  $[8, 15]$ . The total 8 numbers results to  $8!$  permutations while with repetition in a sequence  $S_c$  of size  $n$  the total number of available codes are  $8^n$ . In case where a  $6 \times 6$  plane is utilized consisted of 36 blocks more than  $3 \cdot 10^{32}$  codes are generated. In Figure 4.4 we depict the proposed code-based multiple SiP architecture with the embed and extract procedures in sequential steps.

### **ROMFAIA: Hybrid Watermarking Model for Authenticity and Integrity Preservation**

The proposed framework provides a unified solution for the preservation of both authenticity and integrity of medical images through the sequential deployment of robust and fragile watermarking mechanisms. More specifically, the framework consists of two complementary security layers. The first layer focuses on authenticity preservation through the embedding of a robust code-based watermark representation throughout the image, while the second layer focuses on integrity preservation through cryptographic verification relationships established among image regions.

### **Security Enhancement through HMAC-SHA-256.**

Compared to the previously proposed cross-block integrity-preservation model, which employed the SHA-256 cryptographic hash function, the proposed ROMFAIA framework adopts HMAC-SHA-256 in order to integrate secret-key authentication into the integrity verification procedure. While both approaches generate a 256-bit verification value, the incorporation of a secret key  $K$  significantly strengthens the security of the integrity-preservation mechanism.

More specifically, in a conventional SHA-256 based verification framework, an adversary with knowledge of the verification algorithm and the established permutation-based verification relationships could recompute valid hash values after modifying the watermarked image and subsequently re-embed the forged verification information

---

**Algorithm 4.5** Image Authenticity and Integrity Embedding for the ROMFAIA Model

---

```
1: Input: image  $I$ , code-sequence  $S_c$ , permutation  $P$ , Key  $K$ 
2: Output: image  $I_{rf}$ 
3: Partition  $I$  into a  $d \times d$  authenticity-plane
4: for authenticity-blocks  $B_i$  do
5:   Encode  $S_c(i)$  as a SiP
6:   Construct the corresponding embedding grid
7:   Embed the SiP into  $B_i$  using the FFT-based procedure
8: end for
9: Obtain intermediate watermarked image  $I_r$ 
10: Partition  $I_r$  into a  $k \times k$  integrity-plane
11: for ordered pairs  $(B_p, B_q)$  defined by  $P$  do
12:   Compute  $h \leftarrow \text{HMAC}(B_p^R, K)$ 
13:   Embed  $h$  into the LSBs of  $B_q^G$ 
14: end for
15: Generate image  $I_{rf}$ 
16: Return  $I_{rf}$ 
```

---

into the corresponding *integrity-blocks*. Under such a scenario, the manipulated image could potentially satisfy the integrity verification procedure despite containing unauthorized modifications.

By replacing SHA-256 with HMAC-SHA-256, the proposed framework enhances the security against this vulnerability. Since the integrity information depends on both the image content and the secret authentication key  $K$ , valid verification values cannot be reproduced without knowledge of the secret key, even when the integrity verification algorithm and the permutation-based mapping are fully known. Consequently, only authorized parties possessing  $K$  are able to generate valid integrity information, substantially improving the resistance of the proposed framework against forgery and malicious re-embedding attacks.

### **Embedding Authenticity Information.**

Authenticity preservation is achieved through the deployment of a code-based watermarking scheme that embeds a sequence of integers throughout the image. Following image partitioning, a code-sequence is generated and each integer is encoded as a

Self-Inverting Permutation. The resulting SiP structures determine the embedding locations within each *authenticity-block*.

For every selected embedding region, frequency-domain processing is performed using the Fast Fourier Transform. Watermark information is embedded by modifying magnitude coefficients within predefined annular regions of the Fourier spectrum while preserving the corresponding phase information. The modified frequency components are subsequently transformed back into the spatial domain, producing the intermediate watermarked image containing the embedded authenticity information.

During extraction, the same partitioning and frequency-domain analysis procedures are applied. The embedded SiPs are reconstructed and converted back into their corresponding integer representations, producing an extracted code-sequence. The similarity between the extracted and original sequences is evaluated through the Extraction-Rate metric, which serves as the authenticity validation criterion. To further improve robustness, a reconstruction procedure exploits the structural properties of Self-Inverting Permutations in order to recover partially corrupted watermark representations prior to validation.

### **Embedding Integrity Information.**

Integrity preservation is achieved through cryptographic verification relationships established among *integrity-blocks*. Following the authenticity embedding stage, the image is partitioned into an *integrity-plane*. For each *integrity-block*, a cryptographic digest is generated from the red-channel information utilizing the HMAC-SHA<sub>256</sub> function.

Rather than storing the generated hash within the same block, the framework embeds it into another *integrity-blocks* according to a permutation-based mapping. Consequently, each block participates in a verification relationship in which one block verifies another. By extending these relationships across multiple blocks, the framework constructs chains of interdependent verification structures referred to as  $\Lambda$ -relations.

During verification, hash values are recomputed and compared with the corresponding embedded information. Verification failures indicate that at least one of the involved blocks has been modified. The introduced  $\Lambda$ -relations enable the framework to resolve this ambiguity by analyzing both incoming and outgoing verification dependencies. Blocks that neither successfully verify nor are successfully verified by neighboring blocks are classified as tampered. As a result, the framework provides both image-level integrity assessment and precise localization of manipulated regions.

---

**Algorithm 4.6** Image Authenticity and Integrity Verification for the ROMFAIA Model

---

```
1: Input: Image  $I_{rf}$ , Permutation  $P$ , Key  $K$ 
2: Output:  $R_e$ , detected image  $I_d$ 
3: Partition  $I_{rf}$  into the corresponding integrity-plane
4: for authenticity-block  $B_i$  do
5:   Extract the embedded SiP
6:   Reconstruct the corresponding integer
7: end for
8: Construct the extracted code-sequence  $S_c^e$ 
9: Apply the reconstruction procedure if required
10: Compute the Extraction-Rate  $R_e$ 
11: Partition  $I_{rf}$  into the corresponding integrity-plane
12: for ordered pairs  $(B_p, B_q)$  defined by  $P$  do
13:   Extract the embedded hash from  $B_q^G$ 
14:   Compute  $h \leftarrow HMAC(B_p^R, K)$ 
15:   Compare the extracted and computed hashes
16: end for
17: Analyze the resulting  $\Lambda$ -relations
18: Identify tampered Integrity-blocks
19: Generate the tamper localization map
20: Return  $R_e, I_d$ 
```

---

**Verification and Decision Making.**

The final verification stage combines the outputs of the authenticity and integrity preservation procedures in order to assess the security status of the examined medical image. The authenticity component evaluates the correctness of the extracted code-sequence through the Extraction-Rate metric, determining whether the embedded watermark information remains intact. Simultaneously, the integrity component analyzes the established verification relationships among *integrity-blocks* in order to detect inconsistencies caused by malicious modifications.

The combination of these two complementary mechanisms enables the framework to provide both authenticity validation and integrity verification. Consequently, the

proposed approach not only determines whether a medical image remains authentic but also identifies and localizes potentially manipulated regions whenever integrity violations are detected. The resulting decision provides a comprehensive assessment regarding the trustworthiness and reliability of the protected medical image. The detailed steps of the proposed model are presented in algorithms 4.5 and 4.6.

### **Extension to NFT Image Authentication Through the ProtAIN Model**

Non-Fungible Tokens (or, for short, NFT) establish the ownership and the authenticity of digital assets. Regardless of the high security levels provided by blockchain, the cases where the ownership of a digital object may be claimed without proper rights are not excluded. Fraud and copyright infringement should be prevented through the assurance of the authenticity and the integrity of a digital object associated with an NFT. In this work we consider the case of NFT images under the aspect of being susceptible to duplication, theft, tampering, fraud, or, in general, any potential threat against their authenticity and integrity properties.

To deal with such cases, we extend the principles introduced by our ROMFAIA model – originally proposed for the protection of forensic medical images – to the domain of NFT images. ROMFAIA combines a robust, code-based watermarking layer deployed in the frequency domain for authenticity preservation with a fragile, cryptographically verified watermarking layer deployed in the spatial domain for integrity preservation, enabling both the validation of an image’s origin and the detection and localization of unauthorized modifications. The same dual-layer rationale is highly relevant to NFT images, which face an analogous need to prove that the asset associated with a token has not been altered since minting, while also being able to pinpoint the exact regions of an image that may have been tampered with after the fact.

Building on this foundation, we propose an integrated system that provides an a priori embedding of authenticity- and integrity-related information by deploying robust authenticity information hiding in the frequency domain alongside fragile integrity information hiding in the spatial domain, before the image is made publicly available as an NFT. Moreover, an automated secure procedure for the development of smart contracts is proposed, alongside the development of an NFT dual that serves as a backup and is not publicly available through the marketplace, in order to retain the initial digital image. We design a modular architecture that incorporates all the

components proposed in this work and discuss its potential for deployment over different digital objects that can be considered as NFTs.

### **Security Issues in NFT Images**

There are particular security issues regarding NFT images, considering mostly the ownership and the authenticity, the copyright infringement, metadata exploitation, the vulnerabilities potentially exhibited in the smart contracts, storage vulnerabilities and the Ponzi schemes. Metadata attached to NFT images can contain sensitive information about the creator, owner, or the image itself. If this metadata is not properly managed, it can be exploited by malicious entities for identity theft, or other harmful activities. NFTs are typically implemented using smart contracts on blockchain platforms like Ethereum. Smart contracts are susceptible to bugs and vulnerabilities that can be exploited by attackers to manipulate or steal NFTs. Security audits and thorough testing of smart contracts are essential to mitigate these risks. NFT images are often hosted on centralized servers or decentralized storage platforms. Vulnerabilities in these hosting environments can lead to unauthorized access to the image files, compromising their integrity and security. Malicious actors may use social engineering techniques to trick NFT owners into revealing their private keys or other sensitive information, allowing them to gain unauthorized access to and control over the associated NFTs. The progress of NFTs has attracted scammers who exploit inexperienced investors by selling fake or overvalued NFT images. To mitigate these security issues [54], developers, platforms, and investors should prioritize security best practices such as encryption, secure authentication mechanisms, code audits, and user education on safe practices. Additionally, regulatory frameworks may evolve to address legal and security concerns related to NFTs and digital assets. The vulnerabilities associated with NFTs can be formally outlined as follows:

- **Ownership Disputes:** NFT ownership may be subject to disputes or ambiguity, potentially resulting in conflicts over ownership rights.
- **Metadata Tampering:** There exists a risk of tampering with the metadata associated with NFTs, which can lead to misrepresentation or loss of authenticity.
- **Smart Contract Vulnerabilities:** Vulnerabilities in the smart contracts governing NFTs may be exploited, facilitating unauthorized transfers or manipulation of ownership rights.



- **Platform Security:** NFT marketplaces and platforms are susceptible to various security threats, including hacking, fraud, and phishing attacks, which can result in the compromise or theft of NFTs.
- **Legal and Regulatory Uncertainty:** The evolving legal and regulatory landscape surrounding NFTs introduces uncertainty and potential risks for creators and buyers.

### **Smart Contracts, Blockchain and NFTs**

Smart contracts represent a groundbreaking advancement within the realm of blockchain technology [55]. Initially conceptualized in the 1990s as a digital transaction protocol, smart contracts have evolved into self-executing, autonomous pieces of code that encode and enforce the terms of an agreement without reliance on intermediaries or centralized legal bodies. By leveraging the decentralized nature of blockchain networks, where transactions are automatically executed and verified by network nodes, smart contracts reduce transaction risk and administration costs while offering enhanced security and transparency compared to traditional contract mechanisms [56]. Their applications span finance, supply chain management, decentralized finance (DeFi), and NFT marketplaces, executing predefined actions when specific conditions are met [57]. However, smart contracts also exhibit notable vulnerabilities [58]: code bugs may lead to unauthorized access, manipulation of contract terms, or loss of funds; reentrancy issues can arise when a contract calls an external contract before completing its own state changes, allowing malicious external contracts to re-enter and exploit unvalidated calls to manipulate state or drain funds; and improperly handled arithmetic operations may introduce integer overflow or underflow vulnerabilities. Mitigating these risks requires thorough testing, code audits, standardized libraries, access controls, minimized reliance on external data sources, and ongoing monitoring.

Blockchain itself is a decentralized and distributed digital ledger technology that enables the secure, transparent, and immutable recording of transactions across a network of interconnected nodes, with each transaction grouped into a cryptographically linked "block" to form a chronological chain [59]. By removing the need for a central authority or intermediary, and with each node maintaining a complete, synchronized copy of the ledger, blockchain provides redundancy, resilience, and tamper-proof transaction records secured through cryptographic hashing and consensus mecha-

nisms [60, 61, 62], supporting applications such as financial transactions, supply chain management, and identity verification [56]. NFTs build on this infrastructure as unique, indivisible cryptographic tokens whose ownership and transaction history are securely stored and verified on the blockchain, typically representing ownership or proof of authenticity of digital or physical assets such as artwork, collectibles, or virtual real estate [60]. Since blockchain itself is not designed for bulk file storage, NFT assets are commonly stored on the InterPlanetary File System (IPFS), a decentralized peer-to-peer protocol that replaces centralized hosting with a distributed network of nodes; each file is assigned a unique Content Identifier (CID) based on its content, enabling location-independent retrieval, efficient distribution through replication and caching, and reduced storage overhead through versioning and deduplication [63].

### **Automated Development of Smart Contracts and NFT Duals**

Next, we present the implemented technique for the automated development of Smart Contracts, as also the development of NFT duals that are stored as backup for the protection of NFT images. The embedding and verification of the authenticity and integrity-related information follow the ROMFAIA model.

The primary vulnerabilities associated with NFTs stem from ownership disputes and the conflicts they create over ownership rights, from tampering that compromises authenticity, and from vulnerabilities inherent to smart contracts themselves. To address these issues, this model introduces a method for the automated development of smart contracts together with the construction of an NFT dual, a non-identical copy of the original image that is uploaded to IPFS prior to the image itself, serving as a backup. The automated smart contract generation process was implemented in Python, with a local IPFS daemon used to upload the images and the Ganache provider employed for blockchain testing. Smart contracts, written in Solidity 0.8.9, are generated from the console based on user responses regarding permissions and potential listing on the marketplace; each contract contains the CID of the processed image along with functions primarily related to NFT transactions. The smart contract permissions initially provided by the automated procedure of ProtAIN are listed below:

- Initiate NFT smart contract as ownable only.
- Initiate NFT smart contract as ownable with operator filterer access.

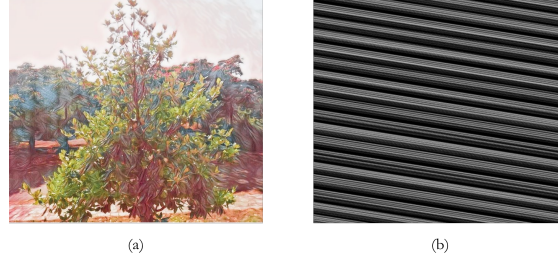


Figure 4.5: Arnold's transformation for the initial image (a) to develop its NFT "Dual" (b) for the ProtAIN model.

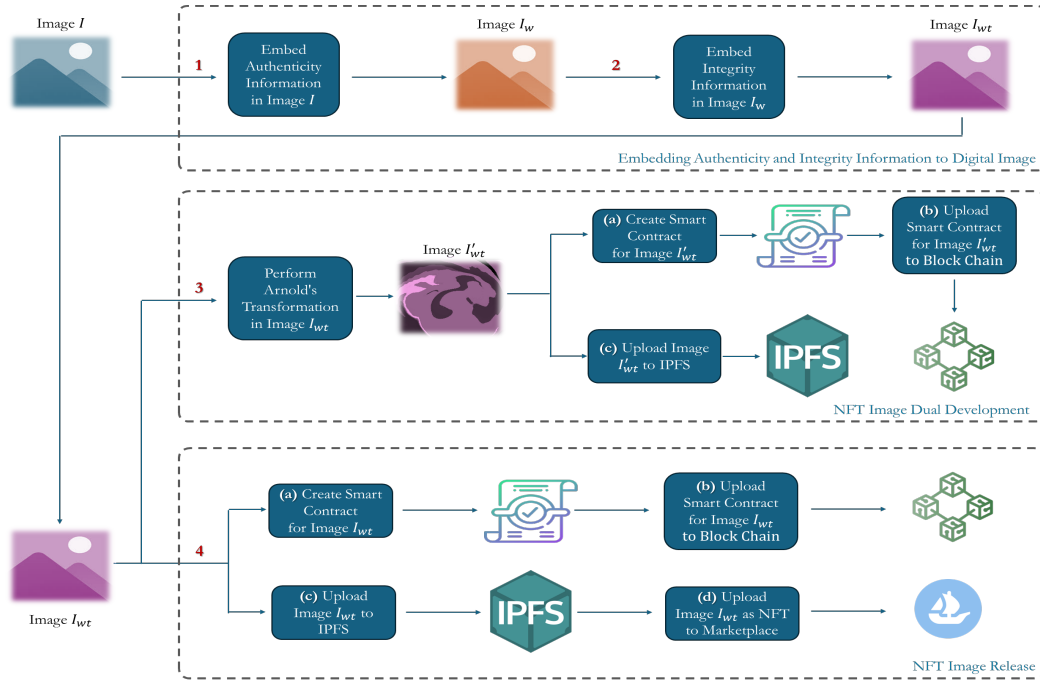


Figure 4.6: ProtAIN system architecture and procedure deployment.

- Write methods owner only about change sale state/set uri/set price.
- Build smart contract for OpenSea NFT sale.

In the proposed scheme, the original image is first uploaded to IPFS to serve as proof of authenticity, after which the corresponding smart contract is executed on the blockchain and its hash is generated. In addition, we introduce the use of an image "dual": given an image  $I$  on which the authenticity and integrity information has been embedded sequentially, resulting in  $I_{wt}$ , we produce its corresponding dual  $I'_{wt}$  by applying a series of Arnold transformations, the number of which corresponds to an integer used as a secret key. The transformed image  $I'_{wt}$  (Figure 4.5 (b)) is uploaded to IPFS together with the creation of its own smart contract before  $I_{wt}$  itself

is uploaded to IPFS and listed on the marketplace. The backup image  $I'_{wt}$  (Figure 4.5 (a)) remains on IPFS but is not made publicly available, so that even if unauthorized access is obtained, the original image cannot be reconstructed without knowledge of the number of Arnold transformations applied. As a result,  $I'_{wt}$  together with its smart contract serves as a future-proof attestation for  $I_{wt}$  and its own smart contract: should an attack occur, the backup formed by  $I'_{wt}$  and its smart contract can serve as evidence to support the authenticity and integrity of the original NFT image. The overall architecture of the ProtAIN model is depicted in Figure 4.6.

### 4.1.2 Audio

In this section we describe the proposed algorithms for audio watermarking and watermark extraction to and from an audio signal. We describe the procedure utilized for embedding watermark information into digital images, as well as the subsequent embedding of the produced watermarked images into audio signals. Furthermore, the corresponding extraction procedure is presented, where the embedded watermarked image is reconstructed from the digital audio signal in order to recover the embedded watermark information and verify the authenticity of the audio file. In Figure 4.7 it is presented an illustrative example of how these procedures are deployed in order to implement our proposed multiple watermarking scheme for digital audio signals. It is worth noting that the basic watermarking approach was selected in the proposed audio watermarking model because the images utilized correspond to logo-sized images with fixed dimensions of  $128 \times 128$  pixels. Consequently, the available cell size within the image is relatively limited, restricting the effective deployment of the repetitive watermarking approach. A more detailed discussion regarding this design choice and its corresponding limitations is provided in Section 7.

#### Embedding Procedure

The proposed dual-layer audio watermarking framework combines the SiP-based image watermarking methodology of [1] with an LSB-based audio watermarking scheme [15, 16]. Initially, an integer watermark  $w$  is encoded into its corresponding SiP and embedded into image  $I$  according to its 2DM representation, producing a watermarked image  $I_w$ . Subsequently, the grayscale image  $I_w$  of dimensions  $m \times n$  is transformed into a one-dimensional vector of length  $\ell = m \times n$ . Since each pixel

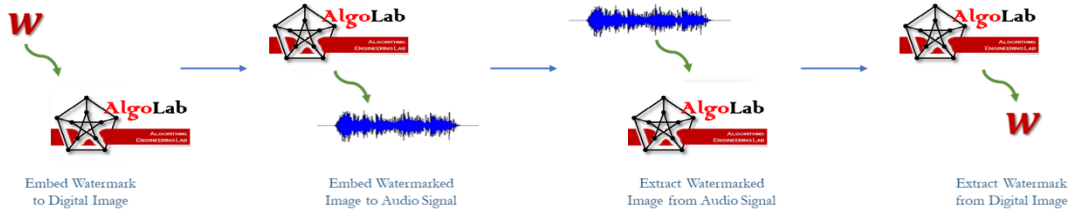


Figure 4.7: Overview of the procedures deployed for the embed and the extract of multiple watermarks into and from digital audio signals.

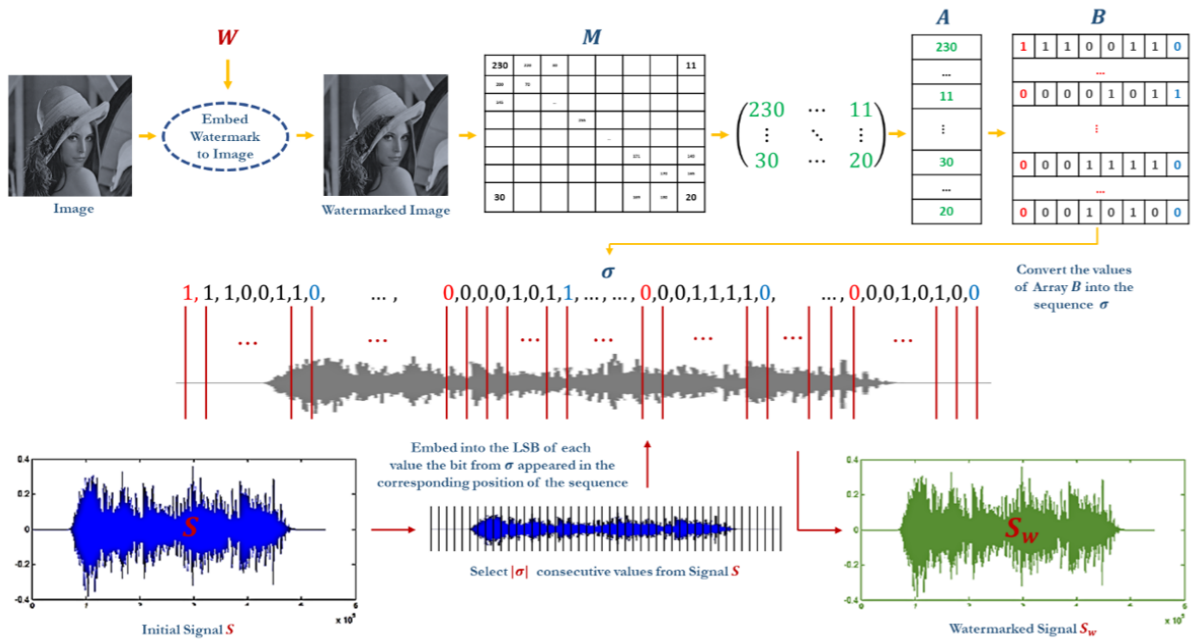


Figure 4.8: Embedding procedure of the integer  $w$  into the image  $I_w$  followed by the embedding of  $I_w$  into the LSB's of the audio signal  $S$ .

value belongs to the range  $[0, 255]$ , it is represented using 8 bits, resulting in a binary sequence  $\sigma$  of length  $\kappa = \ell \times 8$ . Let  $|S|$  denote the number of samples of the host audio signal  $S$ . Using a step size  $step = \frac{|S|}{\kappa}$ , the bits of  $\sigma$  are uniformly embedded into the LSBs of selected *audio-samples*, producing the final watermarked audio signal  $S_w$ . Figure 4.8 illustrates the overall embedding architecture, while Algorithm 4.7 presents the detailed embedding procedure.

### Extraction Procedure

The extraction procedure follows the reverse process. Given a watermarked audio

---

**Algorithm 4.7** Embed Watermarked Image to Audio Procedure

---

```
1: Input: Audio signal  $S$ 
2: Output: Watermarked audio signal  $S_w$ , Image  $I$ 
3: Watermark image  $I$  to produce  $I_w$ 
4:  $S_w \leftarrow S$ 
5: Let  $m, n$  denote the dimensions of image  $I_w$ 
6: Let  $M[i, j]$  denote the pixel located at position  $(i, j)$  of image  $I_w$ 
7: Construct matrix  $A$  of dimensions  $1 \times \ell$ , where  $\ell = m \times n$ 
8:  $counter \leftarrow 0$ 
9: for  $i = 1 : 1 : m$  do
10:   for  $j = 1 : 1 : n$  do
11:      $A[counter++]\leftarrow M[i, j]$ 
12:   end for
13: end for
14: Construct matrix  $B$  of dimensions  $\ell \times 8$  containing the 8-bit binary representation
    of each element of matrix  $A$ 
15: Construct sequence  $\sigma$  of length  $\kappa = \ell \times 8$  by sequentially parsing the rows of
    matrix  $B$ 
16: Select  $\kappa$  audio-samples  $s_1, s_2, \dots, s_k \in S_w$ ,  $k = |\sigma| \leq |S_w|$ 
17: for  $i = 1 : \frac{|S_w|}{\kappa} : |S_w|$  do
18:    $LSB(s_i \in S_w) \leftarrow \sigma_i$ 
19: end for
20: Construct the final watermarked audio signal  $S_w$ 
21: Return  $S_w$ 
```

---

signal  $S_w$  and the dimensions  $m \times n$  of the embedded image,  $\kappa = \ell \times 8$  *audio-samples* are selected using the same step size  $step = \frac{|S|}{\kappa}$ , where  $\ell = m \times n$ . The LSB of each selected *audio-sample* is extracted, producing a binary sequence  $\sigma$  of length  $\kappa$ . The sequence is subsequently partitioned into groups of 8 bits, forming a binary matrix whose rows correspond to the binary representation of grayscale pixel values. By converting each row to its decimal representation, a one-dimensional vector containing the recovered pixel intensities is obtained. Finally, the vector is reshaped into a matrix of dimensions  $m \times n$ , reconstructing the embedded watermarked image  $I_w$ . The overall extraction architecture of the proposed dual-layer audio watermarking

---

**Algorithm 4.8** Extract Watermarked Image from Audio Procedure

---

```
1: Input: Watermarked audio signal  $S_w$ , integers  $m, n$ 
2: Output: Watermarked image  $I_w$  of dimensions  $m \times n$ 
3: Construct an empty sequence  $\sigma$  of length  $m \times n \times 8$ 
4: Select  $\kappa$  audio-samples  $s_1, s_2, \dots, s_k \in S_w$ ,  $k = |\sigma| \leq |S_w|$ 
5: for  $i = 1 : \frac{|S_w|}{\kappa} : |S_w|$  do
6:    $\sigma_i \leftarrow LSB(s_i \in S_w)$ 
7: end for
8: Construct matrix  $B$  of dimensions  $\ell \times 8$ , where  $\ell = m \times n$ 
9: Parse sequence  $\sigma$  and place each group of 8 binary digits into the corresponding
   row of matrix  $B$ 
10: Construct matrix  $A$  of dimensions  $1 \times \ell$ , where each cell contains the integer value
    corresponding to the 8-bit binary representation of the respective row of matrix
     $B$ 
11: Construct matrix  $M$  of dimensions  $m \times n$ 
12:  $counter \leftarrow 0$ 
13: for  $i = 1 : 1 : m$  do
14:   for  $j = 1 : 1 : n$  do
15:      $M[i, j] \leftarrow A[counter++]$ 
16:   end for
17: end for
18: Construct the final watermarked image  $I_w$  from matrix  $M$ 
19: From  $I_w$  extract the embedded integer  $w$ 
20: Return  $I_w, w$ 
```

---

framework is illustrated in Figure 4.9, while Algorithm 4.8 presents the detailed extraction procedure.

### 4.1.3 Video

In this section, the proposed watermarking model for authenticity preservation of digital video objects is presented. The proposed approach combines video and audio information embedding in order to provide both authenticity and integrity verification protecting both visual and auditory components. More specifically, the video frames

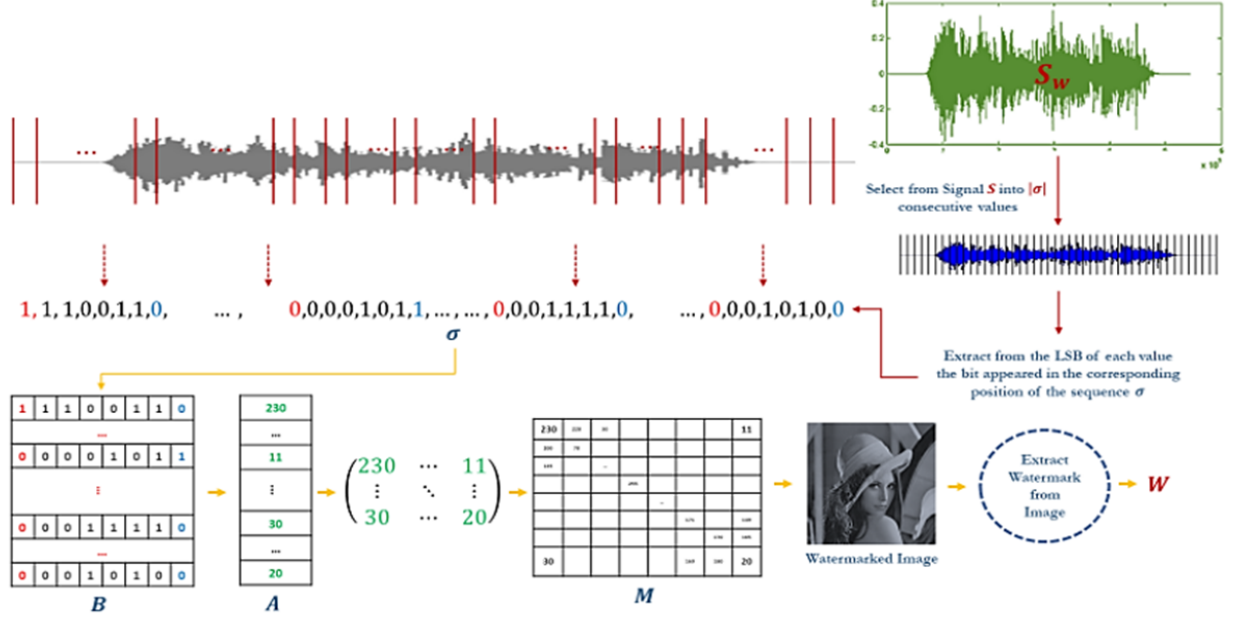


Figure 4.9: Extracting procedure of the watermarked image  $I_w$  from the LSB's of the audio signal  $S_w$  followed by the extraction of the integer  $w$  from  $I_w$ .

are partitioned into non-overlapping *authenticity-blocks*, while watermark information is represented through a sequence of integers encoded as 2DM representations of SiPs. Each *authenticity-block* is further partitioned into *cells*, where the watermark embedding process is applied in the frequency domain through modification of selected high-frequency FFT magnitude coefficients as we described in section 3.5.1. Furthermore, the proposed model incorporates an additional audio-based integrity mechanism where a global hash value is computed from the video frames, and its binary representation is embedded into the LSBs of selected audio *audio-samples*.

### Splitting Video to Components

In order to embed authenticity information into the visual frames and integrity information into the audio stream of the video object, the video is initially decomposed into its corresponding visual and audio frames. Each visual frame is associated with a corresponding audio frame through a one-to-one temporal mapping. Each audio frame consists of multiple *audio-samples* and may correspond to either mono or stereo audio configurations. Finally, the extracted visual and audio frames are stored in separate data structures, enabling independent processing of the authenticity and integrity embedding procedures for the visual and auditory components of the digital



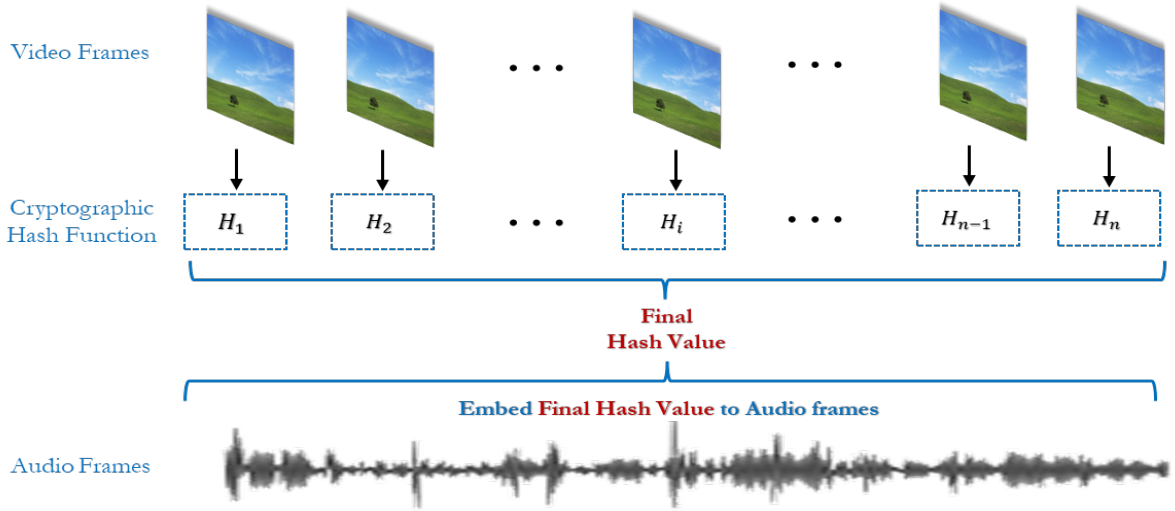


Figure 4.10: Overview of the video-audio binding process. Every video frame is processed by a cryptographic hash function to generate intermediate hash values, which are combined into a final hash. The final hash value is then embedded into the associated audio frames for integrity preservation.

video object.

### Embedding Authenticity Information to Video Frames

Following the code-based multiple watermarking model presented in Section 4.1.1, the proposed video watermarking model utilizes multiple watermark embedding in order to distribute authenticity information across different spatial regions of the video frames. Each video frame is partitioned into *authenticity-blocks*, where each block contains an independent integer watermark represented through the 2DM representation of a SiP. Specifically for each frame we use an  $n \times n$  *authenticity-plane* resulting to  $n^2$  *authenticity-blocks* and an  $k \times k$  *grid*, which results to  $k^2$  *cells*. This configuration is stable in all video frames.

An important advantage of the proposed model is that watermark extraction remains feasible even when parts of the video frames have been corrupted, compressed, removed, or deliberately manipulated. More specifically, regions or frames where watermark extraction fails may indicate potential tampering, while the remaining unaffected spatial and temporal regions can still provide sufficient information for authenticity verification. Unlike single-watermark approaches, where corruption of the embedded region may result in complete watermark loss, the proposed model introduces both spatial and temporal redundancy through the distributed embedding

of the same code-sequence across all video frames. Consequently, the watermark can still be reconstructed even under partial frame degradation or temporal attacks, enhancing the robustness and security of the proposed watermarking model.

### Embedding Integrity Information to Audio Samples

To enable integrity verification throughout the entire lifecycle of the video object, the proposed model constructs a chained hash value that accumulates information from all video frames using the SHA-256 cryptographic hash function. More specifically, for each frame of the video sequence, a unique hash value is computed, while each newly generated value depends on both the current frame and the previously computed hash information. Consequently, the resulting frame-chain produces a final hash value uniquely representing both the visual content and the temporal order of the video frames. Due to the strong sensitivity of hash functions, even minor modifications in individual frames result in significant changes in the generated hash values, enabling efficient detection of frame manipulations.

For a digital video consisting of  $n$  frames, denoted as  $F_1, F_2, \dots, F_n$ , the hash value of each frame is computed as:

$$H_i = \text{hash}(F_i) \quad (4.2)$$

Figure 4.10 illustrates an example of the frame-chain construction process. The generated frame hash values  $H_1, H_2, \dots, H_n$  are concatenated and utilized to compute a final global hash value representing the entire video sequence. The resulting hash value is then embedded into selected LSBs of the corresponding audio stream. The final hash value is computed as follows:

$$H_{\text{final}} = \text{hash}(H_1 \mid H_2 \mid H_3 \mid \dots \mid H_n) \quad (4.3)$$

After computing the final hash value, its binary representation is embedded into the LSBs of the audio stream. More specifically, 256 distinct *audio-samples* are selected, where each sample stores one bit of the binary hash representation. The embedding procedure in the auditory part consists of the following steps:

- Conversion of the final hash value into its binary representation.
- Selection of the corresponding *audio-samples* utilized for the embedding process.

---

**Algorithm 4.9** Video Code-based Hash-Chain Embedding Algorithm

---

**Require:** Video  $V$ , Plane Size  $n$ , Grid Size  $k$ , Integer Sequence  $S_c$

**Ensure:** Watermarked Video  $V_w$

- 1: Split  $V$  into visual frames  $F_1, F_2, \dots, F_n$
  - 2: Split audio stream into corresponding audio frames
  - 3: **for** each visual frame  $F_i$  **do**
  - 4:   Partition  $F_i$  into  $n \times n$  Authenticity Blocks
  - 5:   **for** each Authenticity Block  $B_j$  **do**
  - 6:     Assign corresponding integer watermark  $w \in S_c$
  - 7:     Construct the 2DM representation of  $SiP(w)$
  - 8:     Partition  $B_j$  into a  $k \times k$  grid of cells
  - 9:     **for** each selected cell  $c_{ij}$  defined by the 2DM representation **do**
  - 10:       Compute FFT on  $c_{ij}$
  - 11:       Extract Magnitude  $M_{ij}$  and Phase  $P_{ij}$
  - 12:       Modify selected high-frequency FFT magnitude coefficients
  - 13:       Combine modified  $M_{ij}$  with  $P_{ij}$
  - 14:       Compute inverse FFT on  $c_{ij}$
  - 15:     **end for**
  - 16:   **end for**
  - 17: **end for**
  - 18: **for** each visual frame  $F_i$  **do**
  - 19:   Compute frame hash:  $H_i \leftarrow \text{hash}(F_i)$
  - 20: **end for**
  - 21: Construct final hash:  $H_{\text{final}} \leftarrow \text{hash}(H_1|H_2|\dots|H_n)$
  - 22: Convert  $H_{\text{final}}$  to binary representation
  - 23: Select 256 *audio-samples* from the audio stream
  - 24: **for** each selected audio sample  $A_i$  **do**
  - 25:   Replace the LSB of  $A_i$  with the corresponding hash bit
  - 26: **end for**
  - 27: Reconstruct watermarked audio stream
  - 28: Combine watermarked visual frames and audio stream
  - 29: Export watermarked video  $V_w$
-

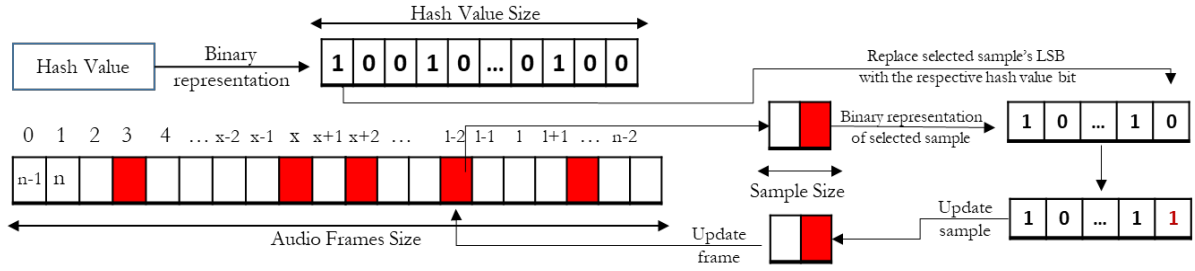


Figure 4.11: Overview of the embedding of the final video-frame hash into selected audio samples through LSB modification.

- Replacement of the LSB of each selected *audio-sample* with the corresponding hash bit.
- Reconstruction of the modified audio stream containing the embedded integrity information.

In the case of mono audio, each selected sample corresponds to a single embedding location. In stereo audio, the corresponding audio channel utilized for embedding must additionally be recorded. The information describing the selected *audio-sample* positions and channel selections is securely stored by the owner in order to enable subsequent extraction and integrity verification. The detailed steps of the embedding procedure are presented in Algorithm 4.9

During the verification procedure, the extraction algorithm also presented in Algorithm 4.10, utilizes the stored embedding information to locate the modified *audio-samples*, extract the embedded bits, reconstruct the final hash value, and compare it with the hash value computed from the received video frames. A successful match between the two hash values indicates that the video content has remained authentic and unmodified, whereas a mismatch indicates potential tampering or manipulation of the video object. Figure 4.11 illustrates an example of the embedding procedure of integrity information into the LSBs of the selected *audio-samples*. In Figure 4.12 we illustrate the overall architecture of the proposed video watermarking model.

### Security Considerations of the Proposed Model

In Table 4.1, we present a comparative security analysis of different frame–audio binding strategies under three representative attack categories, namely geometric attacks, filtering attacks, and audio tampering attacks. More specifically, the analysis

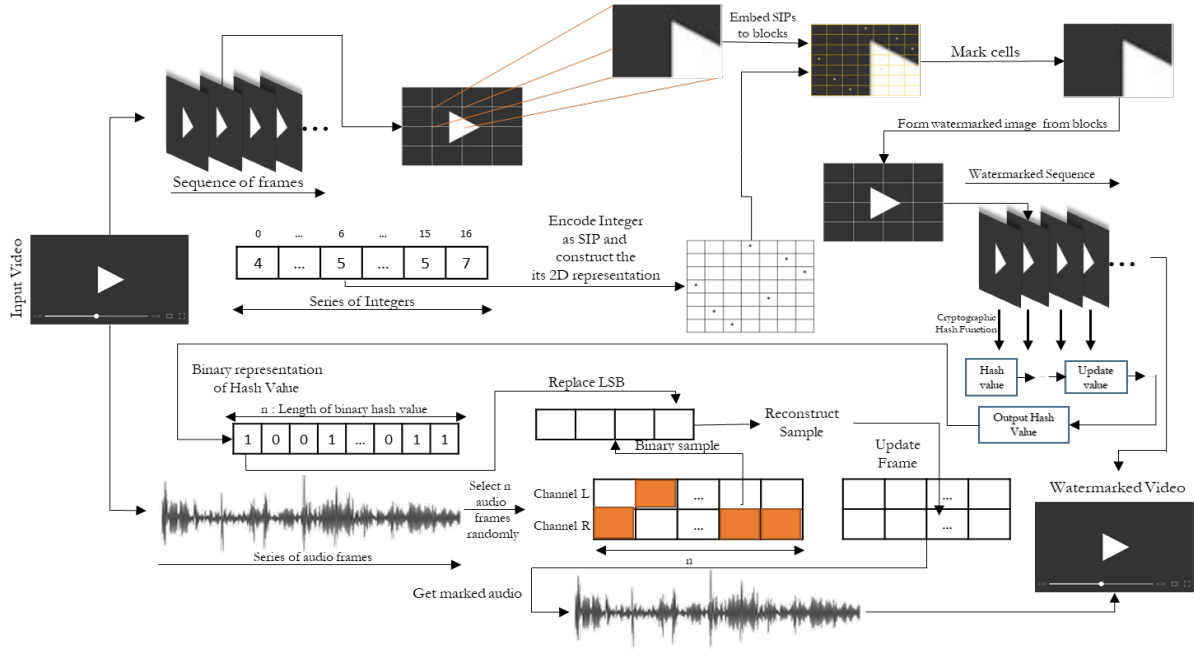


Figure 4.12: Architecture of the proposed video watermarking model.

evaluates the capability of each strategy to preserve the integrity and authenticity of the protected video object under both spatial and temporal manipulations. When only frame-level watermarking is utilized, robustness against localized geometric and filtering distortions may be partially preserved through the embedded frame watermarks. However, the absence of any binding between the visual and auditory components leaves the model vulnerable to audio tampering attacks, since modifications applied to the audio stream do not affect the verification process of the visual watermarks. A one-to-one binding strategy between frame hashes and audio samples introduces a stronger association between video frames and audio data. Nevertheless, such local bindings remain susceptible to temporal manipulations, including frame removal, frame swapping, or partial filtering attacks, since valid hashes may still exist for unaffected frame regions. Similarly, random frame-audio associations provide limited protection, as the lack of a globally dependent integrity representation allows partial inconsistencies between the visual and auditory streams to remain undetected. In contrast, the proposed global frame-chaining strategy constructs a unified integrity representation derived from all video frames and embeds it into the audio stream. Consequently, modifications affecting either the visual or auditory components alter the final hash representation, enabling efficient detection of temporal manipulations while strengthening the overall integrity guarantees of the proposed model.

---

**Algorithm 4.10** Video Code-based Hash-Chain Extracting Algorithm

---

**Require:** Watermarked Video  $V_w$ , Plane Size  $n$ , Grid Size  $k$ , Integer Sequence  $S_c$

**Ensure:** Extracted Watermark Information and Integrity Verification

```
1: Split  $V_w$  into visual frames and audio frames
2: for each visual frame  $F_i$  do
3:   Partition  $F_i$  into  $n \times n$  Authenticity Blocks
4:   for each Authenticity Block  $B_j$  do
5:     Partition  $B_j$  into a  $k \times k$  grid of cells
6:     for each selected cell  $c_{ij}$  do
7:       Compute FFT on  $c_{ij}$ 
8:       Extract Magnitude  $M_{ij}$  and Phase  $P_{ij}$ 
9:       Analyze modified high-frequency FFT coefficients
10:      Recover embedded SiP representation
11:    end for
12:    Decode extracted integer watermark  $w_e$ 
13:  end for
14: end for
15: Extract LSBs from selected audio samples
16: Reconstruct binary representation of:  $H_{\text{final}}^e$ 
17: for each received visual frame  $F_i$  do
18:   Compute:  $H_i \leftarrow \text{hash}(F_i)$ 
19: end for
20: Construct:  $H_{\text{final}} \leftarrow \text{hash}(H_1|H_2|\dots|H_n)$ 
21: if  $H_{\text{final}} = H_{\text{final}}^e$  then
22:   Integrity Verification Successful
23: else
24:   Potential Video Tampering Detected
25: end if
```

---

#### 4.1.4 Documents

In this section we describe the proposed watermarking model for authenticity and integrity preservation of PDF documents. More specifically, we describe the proce-

Table 4.1: Security ablation evaluating the susceptibility of different frame–audio binding strategies to video-level manipulation under geometric, filtering, and audio tampering attacks. GIV stands for Geometric Integrity Vulnerability, FIV for Filtering Integrity Vulnerability and AIV for Audio Integrity Vulnerability.

| Method                                   | GIV        | FIV        | AIV        |
|------------------------------------------|------------|------------|------------|
| Frame Watermarking Only                  | Vulnerable | Vulnerable | Vulnerable |
| Frame Hash → Audio Sample (1–1)          | Vulnerable | Vulnerable | Vulnerable |
| Random-Frame Hash → Random Audio Samples | Vulnerable | Vulnerable | Vulnerable |
| All-Frame Chaining → Audio (Ours)        | Robust     | Robust     | Robust     |

procedure utilized for embedding QR-code watermark information into PDF text objects through spatial-domain modification of text-color values, as well as the corresponding extraction procedure utilized for authenticity verification and tamper detection. The proposed model incorporates Arnold Transformation and permutation-based scrambling mechanisms in order to enhance the security of the embedding and extraction procedures. Furthermore, the QR code is dynamically resized according to the textual content of each PDF page, enabling distributed embedding across the document structure. During the embedding procedure, selected text characters corresponding to black QR-code pixels are slightly modified in their RGB color representation, enabling imperceptible integration of watermark information directly into the PDF document. During the extraction procedure, the embedded QR-code information is reconstructed from the modified text-color values in order to verify the authenticity and integrity of the protected document.

### QR Preprocessing Stages

Prior to the watermark embedding procedure, the proposed PDF watermarking model deploys a preprocessing stage responsible for preparing the QR Code for secure integration into the PDF document structure. Initially, the generated QR Code undergoes grayscale conversion and thresholding in order to produce a binary QR representation suitable for watermark embedding.

Next, the QR Code is resized according to the textual content of the corresponding PDF page, enabling alignment between the QR-code pixels and the available

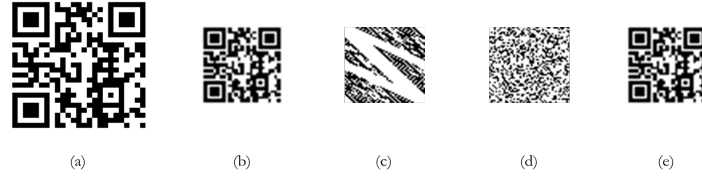


Figure 4.13: Example of the generated QR Code (a), the resized QR Code (b), the QR Code after the Arnold's transformation (c), the QR Code after the permutation rearrangement (d), and the extracted QR Code following the reverse procedure (e).

text-character positions of the document. Figure 4.13 illustrates the preprocessing stages of the QR Code prior to its embedding into the PDF document, including the QR resizing procedure, the sequential deployment of Arnold Transformations, the permutation-based rearrangement process, and the reconstructed QR Code obtained after the inverse extraction procedure.

Following the QR-code preparation stage, two security-oriented preprocessing mechanisms are deployed in order to achieve chaotic diffusion and secure distribution of the watermark information across the document structure.

- **Arnold Transformation.** After the spatial preprocessing stage, the QR Code undergoes Arnold Transformation, where the pixels of the QR are rearranged in a nonlinear and chaotic manner while preserving the original embedded information. Consequently, the transformed QR Code appears visually scrambled and unintelligible to unauthorized users, whereas the original QR Code can only be reconstructed through the corresponding inverse Arnold Transformation. The deployment of multiple Arnold Transformations increases the diffusion of the watermark information and improves robustness against localized attacks targeting specific QR-code regions.
- **Permutation-based Scrambling.** Following the Arnold Transformation, an additional permutation-based scrambling procedure is deployed in order to further diffuse the watermark information across the document structure. More specifically, a permutation equal to the cardinality of the QR-code pixels is utilized to rearrange the transformed QR-code elements prior to the embedding process. The permutation sequence together with the number of Arnold Transformation iterations constitute the security parameters required for successful watermark extraction and QR-code reconstruction.



The combination of Arnold Transformation and permutation-based scrambling significantly increases the security and robustness of the proposed model by diffusing the embedded QR-code information across the entire extent of the document page. Therefore, localized attacks affecting specific document regions are dispersed over multiple areas of the reconstructed QR Code during extraction, reducing the probability of complete watermark corruption. Furthermore, even if an attacker identifies the modified text characters, the extracted embedded image corresponds to the scrambled QR representation, while the attacker remains unaware of both the permutation utilized and the number of Arnold Transformation iterations applied during the embedding procedure. In Figure 4.14 it is provided an illustrative example on the deployment of the baseline method (without the utilization of any pre-processing techniques) over the deployment of the Arnold's transformation, and how this approach augments the robustness of the embedded information against even a targeted attack.

### **Embedding QR-codes to PDFs Text Colors**

Once the QR Code has been prepared for the embedding procedure, the textual content of the PDF document is analyzed character-by-character through a dictionary-iteration process that associates each document character with the corresponding QR-code pixel values (0 or 255). More specifically, the proposed model deploys a dictionary structure containing upper-case and lower-case english alphabet characters in order to identify and map each character of the document to specific QR-code elements. Through this process, each character of the PDF document is associated with a list of QR-code pixel positions, enabling the watermark information to become tightly integrated into the textual structure of the document rather than being directly overlaid as an external graphical object. To further enhance the security of the embedding procedure, the QR-code pixels are embedded according to the permutation sequence generated during the preprocessing stage. The order in which QR-code elements are assigned to text characters is not sequential but permutation-dependent, making unauthorized reconstruction or removal of the watermark significantly more difficult without knowledge of the corresponding permutation key. The actual watermark embedding is then performed through spatial-domain modification of the RGB color values of selected text characters. More specifically, characters corresponding to black QR-code pixels are selected as embedding locations and are assigned an RGB value in the range (1, 0, 0) to (5, 0, 0). This process enables imperceptible watermark

---

**Algorithm 4.11** Embed QR to PDF Algorithm

---

```
1: Input: PDF Document  $D$ , QR Code  $Q$ , Arnold Iterations  $a$ , Permutation  $P$ 
2: Output: Watermarked PDF Document  $D_w$ 
3: Convert  $Q$  to grayscale
4: Apply thresholding on  $Q$ 
5: Resize  $Q$  according to the textual length of each PDF page
6: Apply Arnold Transformation  $a$  times on  $Q$ 
7: Apply permutation-based scrambling utilizing  $P$ 
8: for each page  $p_i$  of  $D$  do
9:   Extract textual content character-by-character
10:  Construct dictionary mapping between characters and QR-code pixels
11:  for each character  $c_i$  of page  $p_i$  do
12:    Associate  $c_i$  with corresponding QR-code pixel value
13:    if QR pixel value = 0 then
14:      Assign RGB value in range (1, 0, 0) to (5, 0, 0)
15:    else
16:      Preserve original RGB value (0, 0, 0)
17:    end if
18:  end for
19: end for
20: Reconstruct PDF document with modified text colors
21: Return Export watermarked PDF document  $D_w$ 
```

---

integration while preserving the readability and perceptual quality of the document. Figure 4.15 presents an overview of the QR reshaping, Arnold scrambling, permutation alignment, dictionary-based character mapping, and character-color embedding procedures utilized in the proposed PDF watermarking model.

Finally, considering that PDF files frequently consist of multiple pages, the embedding procedure is iteratively applied across all pages of the document, enabling distributed watermark embedding throughout the entire PDF structure. After completion of the embedding process, the reconstructed watermarked PDF document is exported preserving both the visual quality of the document and the embedded authenticity and integrity information. The detailed steps of the embedding procedure are presented in Algorithm 4.11.

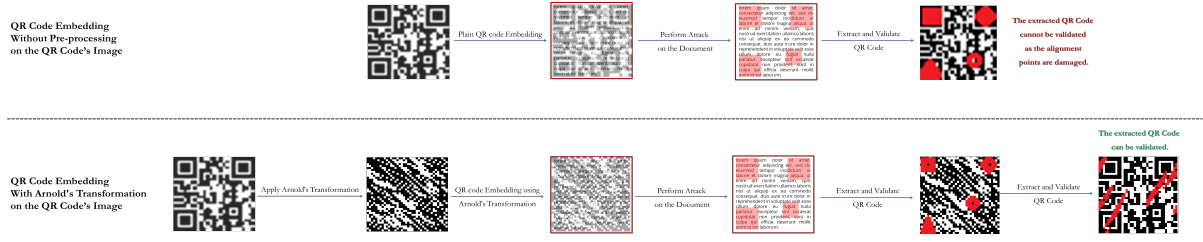


Figure 4.14: Example on the utilization of Arnold's transformation compared to the plain embedding of the QR Code's image against a targeted attack on the document's content.

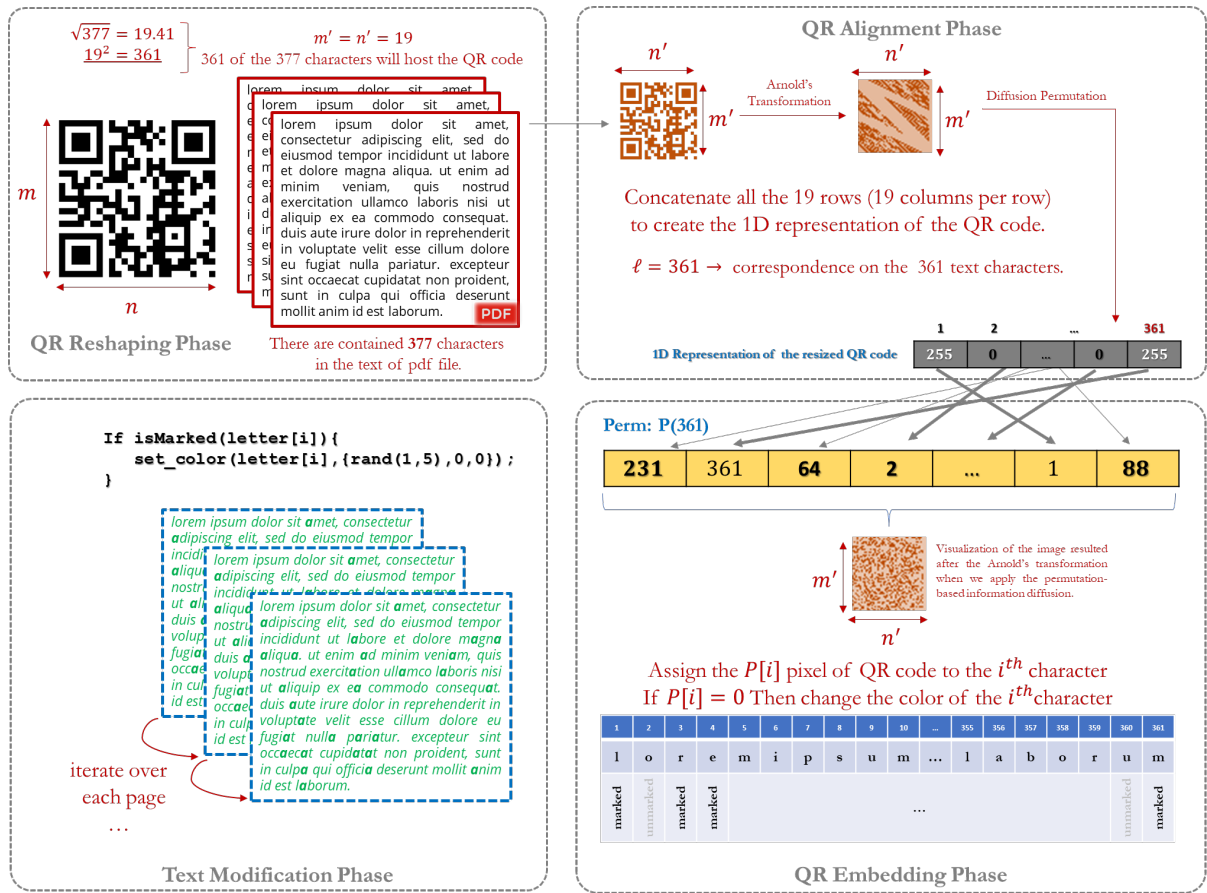


Figure 4.15: Example of the QR Code reshaping and the embedding of the information using Arnold's transformation and permutation to achieve chaotic diffusion of the hidden information.

## Extracting QR-codes from PDFs Text Colors

This section presents the proposed extraction procedure utilized for the recovery and reconstruction of the embedded QR-code from the watermarked PDF document. The

extraction procedure follows the reverse order of the embedding process in order to reconstruct the original QR Code and verify the authenticity and integrity of the protected PDF document. The extraction steps are presented below:

- **Character Color-based Watermark Detection.** Initially, the watermarked PDF document is provided as input to the extraction procedure, which iterates through the textual content of each page and retrieves the RGB color values of every text character. More specifically, characters whose color differs from the default RGB triplet  $(0, 0, 0)$  are identified as potential embedding locations. For every detected modified character, the value 0 is assigned, whereas characters preserving the original color representation are assigned the value 255. Through this procedure, the embedded QR-code information is reconstructed in binary form from the spatial-domain color modifications of the document.
- **Construction of the 1D QR-Code Representation.** Following the character-color analysis, the extracted binary values (0 or 255) are sequentially stored into an one-dimensional array representing the embedded QR-code information. The dimensions of this array correspond to the dimensions of the QR-code utilized during the embedding procedure, enabling reconstruction of the extracted QR-code.
- **Permutation-based Rearrangement of QR Elements.** Since the QR-code information was embedded following the permutation-based scrambling procedure utilized during the preprocessing stage, the extracted QR-code elements must be reordered according to the corresponding secret-key permutation. More specifically, the extraction procedure utilizes the permutation sequence deployed during embedding in order to correctly rearrange the extracted QR-code elements. Without knowledge of these security parameters, unauthorized reconstruction of the QR Code becomes computationally difficult.
- **QR-Code Reshaping.** After the permutation-based reordering procedure, the reconstructed one-dimensional QR-code representation is reshaped into its corresponding two-dimensional form. This process reconstructs the visual structure of the QR Code and enables its recovery and decoding.
- **Inverse Arnold Transformation.** Finally, the inverse Arnold Transformation is applied iteratively in order to relocate the QR-code pixels back to their original

---

**Algorithm 4.12** Extract QR from PDF Algorithm

---

```
1: Input: Watermarked PDF Document  $D_w$ , Arnold Iterations  $a$ , Permutation  $P$ 
2: Output: Reconstructed QR Code  $Q_e$ 
3: for each page  $p_i$  of  $D_w$  do
4:   Extract textual content character-by-character
5:   for each character  $c_i$  of page  $p_i$  do
6:     Retrieve RGB color value of  $c_i$ 
7:     if  $RGB(c_i) \neq (0, 0, 0)$  then
8:       Append value 0 to QR sequence
9:     else
10:      Append value 255 to QR sequence
11:    end if
12:  end for
13: end for
14: Construct the 1D QR-code representation
15: Apply inverse permutation utilizing  $P$ 
16: Reshape the reconstructed sequence to a 2D QR representation
17: Apply inverse Arnold Transformation  $a$  times
18: Return QR Code  $Q_e$ 
```

---

positions prior to the preprocessing stage. Finally, the original QR Code is reconstructed, enabling successful watermark recovery and authenticity verification of the protected PDF document.

The final phase consists of the complete extraction and reconstruction of the QR-code watermark from the watermarked PDF document. Considering that PDF files frequently consist of multiple pages, the extraction procedure iteratively processes all document pages and applies the corresponding watermark-recovery operations independently for each page. Consequently, watermark information distributed throughout the document structure can be effectively reconstructed even in multi-page PDF documents. After completion of the extraction procedure, the reconstructed QR Codes are exported and visualized as image representations for each corresponding PDF page. The detailed of the extraction procedure are present in Algorithm 4.12.

## 4.2 Integrity: Embedding Fragile Information

In this section, the proposed watermarking models developed for the preservation of integrity in images, audio files, PDF documents, and video objects are presented and analyzed. More specifically, the integrity embedding and verification methodologies of each model are described, along with the corresponding mechanisms utilized for tamper detection, localization, and integrity attestation. The proposed approaches are designed with emphasis on security, fragility, and reliability, aiming to efficiently detect malicious modifications while preserving the perceptual quality and structural consistency of the protected digital objects.

### 4.2.1 Images

Simultaneously with the design and implementation of authenticity-preservation techniques, integrity-preservation methods for images were also investigated, with particular emphasis on the detection and localization of malicious manipulations. In this direction, a tamper-detection algorithm was developed utilizing the red and green color channels of the image together with a permutation-based mapping between corresponding *integrity-blocks*. According to the deployed permutation, the cryptographic hash value computed from the red channel of an *integrity-block* is embedded into LSBs of the green channel of another *integrity-block*. Through this process, any modification applied to a block becomes detectable due to both the fragility of the LSB-based embedding procedure and the avalanche-effect property of the cryptographic hash function, where even minimal modifications result in significant changes in the generated hash values. Furthermore, the utilization of the permutation-based inter-block relations enables precise localization of the tampered regions, allowing the proposed framework not only to detect image manipulations but also to accurately identify their spatial position within the forensic image.

#### Embedding Cross-block Integrity Information based on Permutation

Having an image, let  $I$ , of dimensions  $n \times m$ , we define its partitioning into blocks as the grouping of its pixels located into a particular area of the image. More precisely, considering an image of dimensions  $n \times m$ , we divide them by an integer  $k$  producing hence a *grid* of dimensions  $k \times k$  that contains  $k^2$  non-overlapping blocks of equal size. Over each *integrity-block* there are contained  $(n \cdot m)/k^2$  pixels that their values will be

---

**Algorithm 4.13** Cross-block Integrity Information Embedding Algorithm

---

```
1: Input: Image  $I$ , Grid Size  $k$ , Permutation  $P$ 
2: Output: Watermarked Image  $I_{rf}$ 
3: Partition image  $I$  into  $k^2$  non-overlapping integrity-blocks
4: for each ordered pair  $(B_i, B_j)$  defined by permutation  $P$  do
5:   Extract red channel:  $B_i^{\mathcal{R}}$ 
6:   Compute:  $H(B_i^{\mathcal{R}})$ 
7:   Convert hash value to binary representation
8:   Extract:  $B_j^{\mathcal{G}}$ 
9:   Embed the binary hash bits into the LSBs of:  $B_j^{\mathcal{G}}$ 
10: end for
11: Reconstruct image  $I_{rf}$  from modified RGB channels
12: Return  $I_{rf}$ 
```

---

utilized. In particular, the values of the pixels in the red channel of an *integrity-block* are utilized to compute the hash value (in our case, we utilize the SHA-256 hash function), while the LSBs of the pixels in the green channel of another *integrity-block* are utilized to store the bits of the hash value, based on the defined mapping given by the block permutation. Denoting with  $B_i^{\mathcal{R}}$  and  $B_j^{\mathcal{G}}$  the values of the red and the green channel of an *integrity-block*  $B_i$ , in our model it holds:

$$[B_j^{\mathcal{G}}]_{LSB} \leftarrow \text{hash}(B_i^{\mathcal{R}}) \quad (4.4)$$

Considering that the SHA-256 algorithm generates a fixed-length hash representation of 256 bits, each *integrity-block* must contain at least 256 pixels in order to store the complete binary representation of the generated hash value into the corresponding LSB positions. Therefore, the partitioning of the image into *integrity-blocks* is constrained by both the dimensions of the image and the storage capacity required for the embedding of the corresponding cryptographic hash information.

Furthermore, the proposed model establishes inter-block verification dependencies according to the utilized permutation mapping. More specifically, an *integrity-block* may simultaneously act both as a verifier of another *integrity-block* and as a verified block itself. Given three *integrity-blocks*  $B_p \rightarrow B_q \rightarrow B_r$ , the proposed framework defines the corresponding verification dependency as a  $\Lambda$ -relation between the involved *integrity-blocks*. Through these interconnected verification relations, malicious modi-

---

**Algorithm 4.14** Tamper Detection and Localization based on  $\Lambda$ -Relations Algorithm

---

```
1: Input: Watermarked Image  $I_{rf}$ , Grid Size  $k$ , Permutation  $P$ 
2: Output: Tamper Rate  $T_r$ , Highlighted Image  $I_h$ 
3: Define empty list  $L_t$  to store tamper blocks
4: Partition  $I_{rf}$  into  $k^2$  non-overlapping integrity-blocks
5: for each ordered pair  $(B_p, B_q)$  defined by permutation  $P$  do
6:   Extract:  $B_p^R$ 
7:   Recompute:  $H(B_p^R)$ 
8:   Extract embedded hash value from the LSBs of:  $B_q^G$ 
9:   if  $H(B_q^G) \neq H(B_p^R)$  then
10:    Mark  $(B_p, B_q)$  as non-verifiable
11:    if  $B_q$  is successfully verified by another integrity-block then
12:      Continue
13:    else
14:      Mark  $B_q$  as tampered
15:      Store  $B_q$  in  $L_t$ 
16:    end if
17:  end if
18: end for
19: Highlight tampered integrity-blocks constructing  $I_h$  based on blocks stored in  $L_t$ 
20: Compute  $T_r$  as:  $T_r = (|L_t|/TotalImageBlocks) * 100$ 
21: Return  $T_r$  and  $I_h$ 
```

---

fications become difficult to conceal, while the proposed framework enables efficient tamper detection and localization of modified image regions. In Algorithm 4.13 we present the steps of the embedding algorithm.

### Detecting and Localizing Tampering

Given the final watermarked image  $I_{rf}$ , the same *integrity-plane* is used to partition the image into  $k^2$  *integrity-blocks*, and each block is decomposed into its RGB channels. For each ordered pair  $(B_p, B_q)$  defined by the permutation  $P$ , the hash value of the red channel of the *verified* block is recomputed, i.e.,  $H(B_p^R)$ , and compared with the hash value extracted from the LSBs of the green channel of the *verifying* block  $B_q^G$ . A mismatch indicates that tampering has occurred in at least one of the two involved



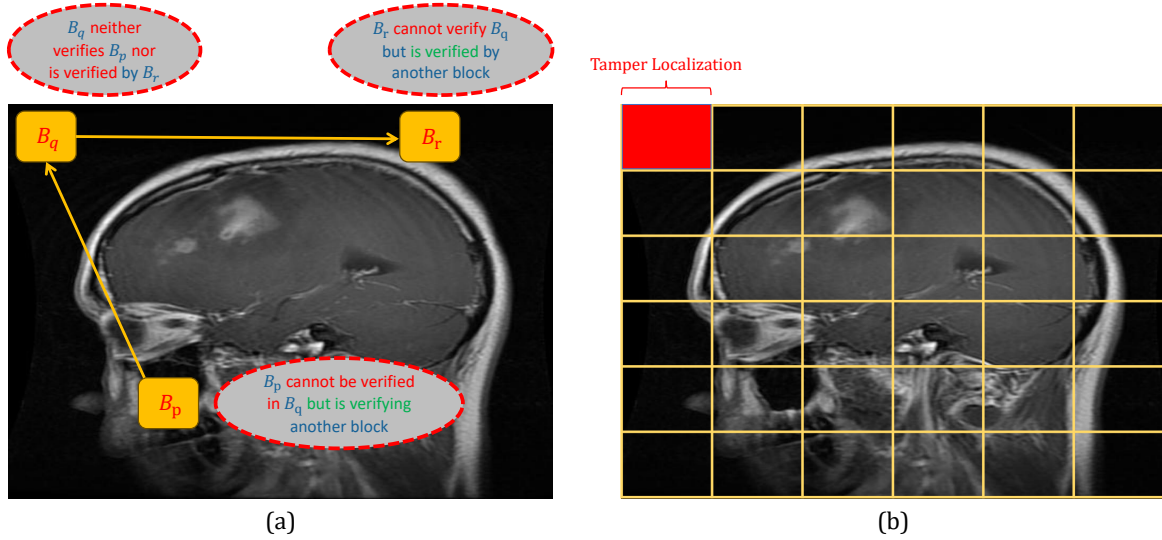


Figure 4.16: Integrity verification and tamper localization. (a)  $\Lambda$ -relation among *Integrity-blocks*, illustrating verification dependencies. (b) Tamper detection and localization based on non-verifiable blocks.

*integrity-blocks*, as illustrated in Figure 4.16(a). At this stage, it is not possible to determine which of the two blocks has been modified. To resolve this uncertainty, we use  $\Lambda$ -relations and we define the following decision rule.

**Tamper Detection Rule:** *An integrity-block is considered tampered if and only if it neither successfully verifies another block nor is successfully verified by another block.*

Based on this rule, blocks that fail both verification directions are marked as tampered (e.g., highlighted in red), enabling precise localization of altered regions, as shown in Figure 4.16(b). After processing all *integrity-blocks*, the method provides a final decision regarding the integrity of the  $I_{rf}$ .

### Cyclic Permutation vs Random Permutation.

A critical component of the proposed integrity-preservation framework is the permutation strategy utilized to establish the verification dependencies between *integrity-blocks*. More specifically, the permutation determines the mapping according to which the hash information computed from an *integrity-block* is embedded into another *integrity-block*. In the proposed framework, both cyclic and random permutation strategies can be utilized, each providing different advantages and limitations re-

garding security, computational complexity, and structural robustness.

In the cyclic permutation strategy, the *integrity-blocks* are connected sequentially in a circular manner, forming a single verification cycle that spans all participating *integrity-blocks*. Unlike randomly generated permutations, cyclic permutations prevent the formation of short verification cycles. For example, under a random permutation it is possible to obtain the pairwise relation  $B_p \rightarrow B_q$  and  $B_q \rightarrow B_p$ , where the hash value of  $B_p$  is stored in  $B_q$  and vice versa. Such two-block verification loops limit the extent of the verification dependencies and reduce the effectiveness of the resulting  $\Lambda$ -relations. In contrast, a cyclic permutation enforces longer verification chains, ensuring that each *integrity-block* verifies a different block while simultaneously being verified by another distinct *integrity-block*. This structure strengthens the inter-block dependencies and improves the reliability of tamper detection and localization. The main limitation of this approach is that the construction of valid cyclic permutations may require additional processing, particularly when large numbers of *integrity-blocks* or more complex verification structures are considered.

Alternatively, the proposed model may utilize random permutation mappings in order to establish the verification dependencies among *integrity-blocks*. In this case, each *integrity-block* is assigned to another *integrity-block* according to a randomly generated permutation sequence. Random permutations provide increased unpredictability and can be generated efficiently with relatively low computational cost, making them attractive for large-scale image processing applications. However, random permutations may produce short verification cycles. For example, it is possible to obtain the relations  $B_p \rightarrow B_q$  and  $B_q \rightarrow B_p$ , where the hash value of the red channel of  $B_p$  is embedded into the green channel of  $B_q$ , while the hash value of the red channel of  $B_q$  is embedded into the green channel of  $B_p$ . In such cases, the verification dependency is confined to only two *integrity-blocks*, limiting the extent of the resulting  $\Lambda$ -relations. Consequently, the verification structure becomes less interconnected compared to longer cyclic verification chains that involve a larger number of *integrity-blocks*. Although random permutations remain suitable for integrity verification, the formation of short cycles may reduce the effectiveness of the inter-block dependency mechanism and the overall strength of the verification relationships.

Consequently, the selection between cyclic and random permutation strategies introduces a trade-off between structural robustness and computational simplicity. Cyclic permutations provide stronger verification consistency by avoiding self-loops,

whereas random permutations provide higher flexibility and lower construction complexity at the cost of potentially weaker verification dependencies in the presence of self-relations.

### 4.2.2 Audio

In parallel with the investigation of integrity-preservation techniques for digital images, a corresponding model for digital audio signals was also developed. The proposed technique embeds integrity-related information directly into the audio utilizing the left and right *audio-channels*. Initially, the audio signal is separated into its corresponding channels and the *audio-samples* are converted into binary form in order to access their LSBs. Next, the samples are partitioned into blocks, while integrity information is generated through XOR operations between the left-channel blocks and a secret key. The produced integrity information is then embedded into the LSBs of the right-channel blocks according to a permutation-based mapping, enabling diffusion of the integrity information across the audio signal. During verification, the corresponding left and right channel blocks are compared according to the deployed permutation and secret-key relations. Any inconsistency indicates potential tampering of the audio signal. Through the combination of XOR-based integrity generation, permutation-driven information diffusion, and fragile LSB embedding, the proposed framework enables efficient tamper detection and localization while preserving the acoustic quality of the protected audio signal.

#### Embedding Integrity Information in Audio

The proposed approach for embedding integrity information in the spatial domain consists of several key stages. Initially, the audio signal is separated into its left and right channels, allowing the corresponding audio *audio-samples* of each channel to be processed independently. Subsequently, the *audio-samples* are converted into binary form in order to enable access and manipulation of their LSBs. The extracted LSBs from both channels are then partitioned into segments, facilitating efficient integrity-information management and controlled embedding operations.

Following this process, integrity information is generated through XOR operations between the LSB segments of the left channel and a secret key. It is worth

---

**Algorithm 4.15** Embedding Integrity Information into Audio

---

```
1: Input: Audio Signal  $S$ , Secret Key  $K$ , Permutation  $P$ 
2: Output: Watermarked Audio Signal  $S_w$ 
3: Separate  $S$  into left and right audio channels
4: Convert audio samples into binary form
5: Extract the LSB sequences from both audio channels
6: Partition the LSB sequences into audio blocks
7: for each audio block  $B_i$  do
8:   Generate integrity information:  $B_i \oplus K$ 
9:   Determine corresponding verifier block:  $B_{P(i)}$ 
10:  Embed the generated integrity information into the LSBs of the right-channel
    block  $B_{P(i)}$ 
11: end for
12: Reconstruct modified audio samples
13: Combine left and right channels constructing  $S_w$ 
14: Return  $S_w$ 
```

---

noting that the XOR operation was selected due to its reversibility property. More specifically, when two binary sequences are combined through an XOR operation, the original information can be reconstructed by performing the XOR operation again between the produced result and either one of the initial input sequences. The resulting integrity-related information is subsequently embedded into the LSB segments of the right channel according to a permutation-based mapping. Through this permutation-driven diffusion mechanism, the integrity information is distributed across the audio signal, increasing both security and robustness against accidental or malicious modifications.

### Audio File Segmentation and Key Generation

To construct the required integrity information, a secret key  $K$  is first generated from the content of the left audio channel. More specifically, the left-channel binary sequence is partitioned into a set of fixed-size chunks denoted as  $C_1, C_2, \dots, C_n$ , where in the proposed implementation  $n = 31$ . Subsequently, an integer watermark  $w$  is selected and encoded as a Self-Inverting Permutation  $SIP_w$  of length 31. The generated permutation determines the order in which the audio chunks participate in the

---

**Algorithm 4.16** Audio Integrity Information Embedding - Secret Key Generation

---

- 1: **Input:** Left-channel Audio Sequence  $L$ , Initial Key  $K_0$ , Integer  $w$
  - 2: **Output:** Secret Key  $K$
  - 3: Partition the left-channel binary sequence into chunks:  $C_1, C_2, \dots, C_n$
  - 4: Generate the corresponding Self-inverting Permutation:  $SIP_w \leftarrow Encode(w)$
  - 5: **for**  $i = 1$  to  $n$  **do**
  - 6:   Select chunk:  $C_{SIP_w(i)}$
  - 7:   Compute:  $H_i = HMAC(C_{SIP_w(i)}, K_0)$
  - 8: **end for**
  - 9: Concatenate the generated hash values according to the permutation ordering:  
     $K = H_1 \circ H_2 \circ \dots \circ H_n \circ K_0$
  - 10: **Return**  $K$
- 

key-generation process.

Next, each chunk is processed using the HMAC-SHA<sub>256</sub> cryptographic hash function and an initial secret key  $K_0$ . Rather than concatenating the resulting hash values according to the original chunk ordering, the proposed framework rearranges them according to the ordering defined by the corresponding  $SIP_w$ . The final secret key is therefore constructed as

$$K = H_{SIP_w(1)} \circ H_{SIP_w(2)} \circ \dots \circ H_{SIP_w(n)} \circ K_0, \quad (4.5)$$

where  $\circ$  denotes the concatenation operation and  $H_{SIP_w(i)} = HMAC(C_{SIP_w(i)}, K_0)$ .

The utilization of the SiP introduces an additional permutation layer into the key-generation process, making the final key dependent not only on the audio content but also on the selected permutation structure. Furthermore, the inclusion of the initial secret key  $K_0$  in the final concatenation step increases both the size and the entropy of the generated key. Since each HMAC-SHA<sub>256</sub> operation produces a 256-bit digest and the proposed implementation utilizes 31 chunks together with the initial key  $K_0$ , the resulting secret key contains  $32 \times 256$  bits. Algorithm 4.16 presents the detailed steps of the key-generation procedure.

### Diffusing Integrity-Information among Audio Channels

After generating the integrity information through the XOR operations with the secret key  $K$ , the resulting binary sequences are embedded into the LSB subsequence of

the right audio channel. To establish the required verification dependencies, the proposed framework utilizes a permutation-based mapping  $P$  among the audio blocks, following the same principle adopted in the image integrity-preservation framework. Through these permutation-defined relationships, the generated verification information is distributed across different blocks and channels, enabling reliable tamper detection and localization while strengthening the overall integrity-preservation mechanism. In Algorithm 4.15 we present the detailed steps of the embedding procedure, including the separation of the audio channels, the extraction and processing of the LSB sequences, the generation of integrity information through XOR operations with the secret key, and the permutation-based diffusion of the integrity information prior to the construction of the final watermarked audio signal.

### **Detecting Malicious Audio Manipulations**

The proposed tamper detection and localization mechanism verifies the integrity relations established between the left and right channel blocks according to the deployed permutation mapping and the XOR-based integrity generation process involving the secret key  $K$ .

During the verification procedure, the integrity information generated from the left-channel blocks is recomputed and compared with the integrity information extracted from the corresponding right-channel blocks. If the recomputed XOR-based integrity information does not match the extracted integrity information, then a verification inconsistency is detected indicating potential tampering of the audio signal. Following the same decision mechanism employed in the image integrity-preservation framework, the proposed audio model utilizes permutation-based verification dependencies among the audio blocks. During verification, inconsistencies in the corresponding integrity relationships indicate the presence of tampering. As a result, modifications affecting the audio samples disrupt the established verification structure, enabling reliable detection and localization of manipulated audio regions.

Finally, the proposed framework preserves the acoustic quality of the protected audio signal, since all modifications remain restricted to the LSB level and therefore remain imperceptible to the human auditory system. Algorithm 4.17 presents the detailed steps of the embedding procedure

---

**Algorithm 4.17** Tamper Detection and Localization in Audio

---

```
1: Input: Watermarked Audio Signal  $S_w$ , Secret Key  $K$ , Permutation  $P$ 
2: Output: Tamper Rate  $T_r$ , Tampered Audio Blocks  $L_t$ 
3: Define empty list  $L_t$ 
4: Separate  $S_w$  into left and right audio channels
5: Convert audio audio-samples into binary form
6: Extract the LSB sequences from both audio channels
7: Partition the LSB sequences into audio blocks
8: for each audio block  $B_i$  do
9:   Recompute integrity information:  $B_i \oplus K$ 
10:  Extract integrity information from verifier block:  $B_{P(i)}$ 
11:  if Recomputed integrity information does not match the extracted information
      then
12:    Mark relation  $(B_i, B_{P(i)})$  as non-verifiable
13:    if  $B_{P(i)}$  is not verified by another related audio block then
14:      Mark  $B_{P(i)}$  as tampered
15:      Store  $B_{P(i)}$  in  $L_t$ 
16:    end if
17:  end if
18: end for
19: Compute:  $T_r = (|L_t|/Total\ Audio\ Blocks) \times 100$ 
20: Return  $T_r$  and  $L_t$ 
```

---

### 4.2.3 Videos

The widespread deployment of CCTV and surveillance systems has increased the need for efficient integrity-preservation techniques capable of protecting digital video evidence from malicious manipulations. Considering a CCTV surveillance scenario as an example, unauthorized modifications applied to video frames or audio streams may compromise the reliability of forensic evidence and security monitoring procedures. In this direction, in the proposed research we introduce a tamper detection and localization model for surveillance videos based on cryptographic hashing and fragile LSB embedding. More specifically, video frames are partitioned into non-overlapping *integrity-blocks*, where hash values computed from the red channel of each *integrity-block* are embedded into the green channel of another *integrity-block* according to a

cyclic permutation mapping. During verification, interconnected  $\Lambda$ -relations between the permuted *integrity-blocks* enable efficient tamper detection and localization. Furthermore, audio-derived hash information is embedded into the corresponding video frames, establishing synchronized audio–visual integrity relations and preventing isolated manipulations of either modality.

### Choosing Cyclic Permutation

In the proposed model, a permutation is constructed over the total set of *integrity-blocks* across all video frames, establishing distributed integrity dependencies throughout the video sequence. The permutation is generated in a one-to-one manner, ensuring that each *integrity-block* is uniquely associated with another *integrity-block*. To strengthen the resulting verification structure, a cyclic permutation strategy is employed, forcing all *integrity-blocks* to participate in a single verification cycle. Unlike arbitrary permutations, which may generate short verification cycles involving only a small number of *integrity-blocks*, the cyclic construction establishes long verification chains spanning the entire set of participating *integrity-blocks*. Consequently, the hash value computed from the red channel of an *integrity-block* is always embedded into the green channel of another *integrity-block*, creating both inter-block and inter-frame verification dependencies. As a result, malicious modifications applied to individual video regions disrupt the corresponding verification chains and propagate through the affected integrity relations, enabling reliable tamper detection and localization while strengthening the overall robustness and security of the proposed video integrity-preservation framework.

### Partitioning Frames and Establishing Cross-Frame Integrity Relations

Consider a video of duration  $d$  seconds and frame rate  $k$ , resulting in a total of  $F = k \cdot d$  frames. Let the spatial resolution of each frame be  $M \times N$ . Each frame is partitioned into a grid of  $B_1 \times B_2$  non-overlapping *integrity-blocks*, where each *integrity-block* has dimensions  $\frac{M}{B_1} \times \frac{N}{B_2}$ . Furthermore, each *integrity-block* must contain at least 256 pixels in order to accommodate the 256 bits produced by the SHA-256 cryptographic hash function. Consequently, a typical *integrity-block* size corresponds to  $16 \times 16$  pixels, although the exact configuration depends on the video resolution.

Subsequently, a cyclic permutation is constructed over the complete set of *integrity-*



---

**Algorithm 4.18** Embedding Cross-modal Integrity Information in Visual and Audio Frames

---

```
1: Input: Video  $V$ , Grid Size  $(B_1, B_2)$ , Cyclic Permutation  $\pi$ 
2: Output: Watermarked Video  $V_w$ 
3: Extract visual and audio frames
4: Partition each visual frame into  $B_1 \times B_2$  non-overlapping integrity-blocks
5: for each integrity-block  $B_i$  do
6:   Compute:  $h_i = H(B_i^R)$ 
7:   Determine corresponding verifier integrity-block:  $B_{\pi(i)}$ 
8:   Embed  $h_i$  into the LSBs of:  $B_{\pi(i)}^G$ 
9: end for
10: for each visual frame  $F_i$  do
11:   Compute hash value of the corresponding audio-frame
12:   Convert the generated hash value into binary form
13:   Embed the binary sequence into the LSBs of the blue channel of frame  $F_i$ 
14: end for
15: Reconstruct watermarked video frames and audio stream
16: Construct watermarked video  $V_w$ 
17: Return  $V_w$ 
```

---

*blocks* across all video frames. For each *integrity-block*, a cryptographic hash value is computed from its red-channel content and embedded into the LSBs of the green channel of another permuted *integrity-block*. In this manner, the integrity information generated from a given *integrity-block* is stored in a different *integrity-block*, potentially belonging to a different video frame.

The utilization of a cyclic permutation ensures that all *integrity-blocks* participate in a single verification chain, avoiding the formation of isolated or short verification cycles. As a result, integrity dependencies are established both within and across video frames, creating a highly interconnected verification structure. During verification, the embedded integrity information is extracted and compared with the recomputed hash values. Any mismatch indicates potential tampering of the corresponding *integrity-block*. To further strengthen the verification process and reduce false-positive detections, the proposed framework additionally exploits chaining-based  $\Lambda$ -relations among the permuted *integrity-blocks*. Finally, detected manipulations are highlighted

---

**Algorithm 4.19** Cross-modal Tamper Detection and Localization in Video

---

```
1: Input: Watermarked Video  $V_w$ , Cyclic Permutation  $\pi$ 
2: Output: Tampered integrity-blocks  $L_t$ , Tampered audio-frames  $L_a$ 
3: Define empty lists  $L_t$  and  $L_a$ 
4: Extract visual and audio frames
5: Partition each frame into non-overlapping integrity-blocks
6: for each integrity-block  $B_i$  do
7:   Recompute:  $h_i = H(B_i^R)$ 
8:   Extract integrity information from:  $B_{\pi(i)}^G$ 
9:   if Extracted integrity information  $\neq h_i$  then
10:    Mark  $(B_i, B_{\pi(i)})$  as non-verifiable
11:    if  $B_{\pi(i)}$  is not verified by another related integrity-block then
12:      Mark  $B_{\pi(i)}$  as tampered
13:      Store  $B_{\pi(i)}$  in  $L_t$ 
14:    end if
15:  end if
16: end for
17: for each visual frame  $F_i$  do
18:   Recompute hash value of the corresponding audio-frame
19:   Extract embedded integrity information from the blue channel of  $F_i$ 
20:   if Extracted integrity information does not match the recomputed hash then
21:     Mark the corresponding audio-frame as tampered
22:     Store the corresponding audio-frame in  $L_a$ 
23:   end if
24: end for
25: Highlight tampered integrity-blocks in red
26: Mute tampered audio-frames
27: Return  $L_t$  and  $L_a$ 
```

---

within the corresponding video regions, enabling direct localization of the affected areas.

### Embedding and Verification of Audio Integrity Information in Visual Frames

Initially, the audio stream is extracted, where each *audio-sample* may consist of either

a single value (mono) or a pair of values (stereo). Subsequently, the audio signal is partitioned into *audio-frames* and a one-to-one correspondence is established between the *audio-frames* and the visual frames of the video sequence, enabling synchronized audio–visual alignment. Leveraging this correspondence, the proposed model extends the integrity-preservation process by incorporating audio-derived integrity information into the visual frames. For each visual frame, the corresponding *audio-frame* is hashed and the resulting binary sequence is embedded into LSBs of the blue channel of the associated visual frame.

During the verification procedure, the hash value of the corresponding audio segment is recomputed and compared with the integrity information extracted from the blue channel of the associated visual frame. Any mismatch indicates potential tampering affecting either the audio or the visual modality. In order to visualize the detected manipulations, the corresponding tampered audio segments may additionally be muted by setting their amplitudes to zero, enabling direct localization of compromised audio regions within the protected video sequence. The embedding procedure steps are presented in Algorithm 4.18, while the verification steps are presented in Algorithm 4.19.

#### 4.2.4 Documents

##### QR-Code Decodability as an Integrity Indicator

An additional integrity-verification mechanism is inherently provided by the embedded QR Code itself. Since the QR Code is embedded directly into the spatial domain through imperceptible modifications of selected text-character colors, its successful reconstruction depends on the preservation of these embedded values. Consequently, unauthorized modifications affecting the watermark PDF document are expected to corrupt the extracted QR Code, potentially rendering it unreadable or non-decodable.

Unlike conventional robust watermarking approaches, where partial degradation of the embedded information may still permit successful watermark recovery, the proposed framework exploits the decoding capability of the QR Code as an additional integrity indicator. More specifically, if the reconstructed QR Code cannot be successfully decoded by a standard QR reader, the document is immediately considered compromised, triggering a tamper alarm. Therefore, the proposed model provides two complementary integrity-verification mechanisms: the successful reconstruction

and decoding of the embedded QR Code verifies the authenticity of the document, whereas failure to decode the extracted QR Code serves as a direct indication of unauthorized modification.

# CHAPTER 5

## ARCHITECTURE: INTEGRATION AND SECURITY

---

### 5.1 Integration Procedure

### 5.2 Image Authenticity and Integrity

### 5.3 Audio Authenticity and Integrity

### 5.4 Video Authenticity and Integrity

### 5.5 Document Authenticity and Integrity

---

### 5.1 Integration Procedure

In this section, we present the architecture of the proposed models and discuss the integration procedures adopted for their implementation. Furthermore, the functionality and interactions of the individual modules composing the proposed systems are analyzed. A summary of the proposed authenticity and integrity preservation frameworks and their main architectural characteristics is depicted in Table 5.1.

### 5.2 Image Authenticity and Integrity

The image authenticity and integrity preservation frameworks developed throughout this dissertation share a common security architecture. More specifically, the proposed methodologies operate on image regions that are utilized for the generation,

Table 5.1: Summary of the proposed authenticity and integrity preservation frameworks and their main architectural characteristics.

| Framework                       | Segmentation Unit                    | Security Information                                            | Embedding Domain                | Verification Mechanism                                         |
|---------------------------------|--------------------------------------|-----------------------------------------------------------------|---------------------------------|----------------------------------------------------------------|
| Repetitive Image Watermarking   | Image Blocks, Cells                  | SiPs                                                            | Frequency Domain                | Watermark Extraction and SiP Reconstruction                    |
| Code-Based Image Watermarking   | Image Blocks, Cells                  | Code-Based SiP Representations                                  | Frequency Domain                | Code Extraction and Authentication Verification                |
| Image Integrity Preservation    | Integrity Blocks                     | SHA-256 Hashes, Permutations, $\Lambda$ -Relations              | Spatial Domain                  | Hash Verification and Tamper Localization                      |
| Audio Authenticity Preservation | Audio Blocks                         | Frequency-Domain Watermarked Images                             | LSB Domain                      | Watermark Recovery and Authentication Verification             |
| Audio Integrity Preservation    | Audio Blocks                         | HMAC-SHA256 Digests, XOR Verification Codes                     | LSB Domain                      | Cryptographic Consistency Verification and Tamper Localization |
| Video Authenticity Preservation | Frames, Audio Segments               | Code-Based SiPs, Global Hash-Chain Representation               | Frequency Domain and Audio LSBs | Watermark Extraction and Frame-Audio Verification              |
| Video Integrity Preservation    | Frames, Frame Blocks, Audio Segments | Cross-Frame Permutations, Cross-Block Relations, Audio Bindings | Spatial Domain and Audio LSBs   | Object-Level and Temporal Tamper Detection and Localization    |
| PDF Authenticity Preservation   | Pages, Document Objects              | QR Codes, Arnold Transformations, Permutations                  | Text Color Modifications        | QR-Code Recovery and Authenticity Verification                 |

embedding, extraction, and verification of authenticity and integrity information. Depending on the selected framework, the generated security structures may consist of watermark representations, cryptographic verification information, or inter-block relationships. The overall security workflow is organized into five interconnected stages, namely object segmentation, security information generation, security information embedding, security information extraction, and security verification and decision making. The interaction among these stages enables the protection, verification, and assessment of image authenticity and integrity.

- **Object Segmentation:** The input image is partitioned into non-overlapping regions (*authenticity-blocks* or *integrity-blocks*) according to the requirements of the selected authenticity or integrity preservation framework. The resulting image blocks constitute the basic processing units utilized throughout the security architecture. In the repetitive and code-based watermarking models, each image block may be further subdivided into a set of embedding cells according to the dimensions of the corresponding SiP. Consequently, the segmentation module can be recursively invoked whenever finer embedding granularity is required. The generated blocks and cells are subsequently forwarded to the security information generation module.
- **Security Information Generation:** Using the image blocks produced during the segmentation stage, the framework generates the security information required for authenticity or integrity preservation. Depending on the selected methodology, this process may involve the generation of watermark structures, cryptographic verification information, and verification dependencies among image blocks. The resulting information, which may include SiPs, code-based SiP representations, cryptographic hash values, and permutation-based verification

relationships. The generated information is subsequently forwarded to the embedding module for insertion into the host image.

- **Security Information Embedding:** The generated security information is embedded into the host image. Depending on the selected framework, embedding may be performed in either the spatial or frequency domain while preserving the visual quality of the protected image and maintaining the required security properties. The output of this module is the protected image object containing the embedded authenticity or integrity information.
- **Security Information Extraction and Verification:** The watermarked image is analyzed in order to retrieve the embedded security information and assess its authenticity or integrity status. Depending on the selected framework, the extracted information may consist of watermark representations, cryptographic hash values, authentication codes, or inter-block verification information. For authenticity-preservation frameworks, the recovered watermark information is compared against the expected watermark representation in order to determine whether the image remains authentic. For integrity-preservation frameworks, the extracted verification information is utilized to identify inconsistencies among image blocks and determine whether malicious modifications have occurred. When applicable, the framework additionally localizes the manipulated regions and produces the final authenticity or integrity decision.

### 5.3 Audio Authenticity and Integrity

The proposed audio authenticity and integrity preservation frameworks share a common architecture that operates on segmented audio signals. More specifically, the audio signal is partitioned into non-overlapping audio blocks that serve as the fundamental processing units for authenticity verification, integrity assessment, and tamper localization. Depending on the selected model, watermark information or cryptographic verification data is generated from the audio content and subsequently embedded within the signal. The common processing pipeline consists of four stages, namely audio segmentation, information generation, information embedding, and information extraction, verification, and decision making. Together, these stages provide

a unified framework for protecting the authenticity, integrity, and security of digital audio signals.

- **Audio Segmentation:** The input audio signal is partitioned into non-overlapping audio blocks that constitute the basic processing units of the proposed frameworks. These blocks are subsequently utilized for watermark embedding, authentication information generation, integrity verification, tamper detection, and localization procedures. The generated audio blocks are then forwarded to the security information generation module.
- **Security Information Generation:** Using the generated audio blocks, the framework constructs the information required for authenticity or integrity preservation. For authenticity-preservation frameworks, this stage may involve the generation of watermark information, including the construction of watermarked images in the dual-layer watermarking model. For integrity-preservation frameworks, cryptographic hash values, HMAC-based authentication information, secret keys, XOR-based verification information, and inter-block verification relationships are generated. The resulting information is subsequently forwarded to the embedding module.
- **Security Information Embedding:** The generated information is embedded into the host audio signal through the corresponding watermarking or integrity-preservation procedure. Depending on the selected framework, the embedding process may involve LSB-based insertion of watermark information, image-to-audio embedding, or the insertion of integrity verification information into selected audio blocks. The resulting watermarked audio signal can subsequently be stored, transmitted, or distributed.
- **Security Information Extraction, Verification, and Decision Making:** The embedded information is recovered from the watermarked audio signal through the corresponding extraction procedure and subsequently utilized to determine the authenticity or integrity status of the examined audio signal. For authenticity-preservation frameworks, the embedded image is reconstructed from the audio signal and the corresponding watermark information is extracted and verified in order to assess authenticity. For integrity-preservation frameworks, the embedded verification information is recovered and compared



against the recomputed verification information generated from the examined audio blocks. Any inconsistency indicates potential tampering of the corresponding audio regions. When applicable, the framework additionally localizes the manipulated audio blocks and produces the final authenticity or integrity decision.

## 5.4 Video Authenticity and Integrity

The proposed video authenticity and integrity preservation frameworks jointly exploit the visual and auditory components of a video object in order to establish authenticity and integrity verification mechanisms. More specifically, the developed methodologies operate on video frames, frame blocks, and audio segments, enabling the generation of watermark information, cryptographic verification data, frame-to-audio bindings, and cross-frame integrity dependencies. The common processing pipeline consists of four stages, namely video segmentation, information generation, information embedding, and information extraction, verification, and decision making. Through the interaction of these stages, the framework supports authenticity verification, tamper detection, tamper localization, and temporal manipulation analysis.

- **Video Segmentation:** The input video object is decomposed into its constituent video and audio streams. The video stream is partitioned into frames and, when required, into frame blocks, while the audio stream is partitioned into audio segments. These processing units constitute the host objects utilized during watermark embedding, hash generation, integrity verification, and temporal manipulation detection procedures.
- **Security Information Generation:** Using the extracted video frames, frame blocks, and audio segments, the framework generates the information required for authenticity or integrity preservation. For authenticity-preservation models, this stage may include the generation of code-based watermark information, cryptographic frame hashes, and frame-to-audio verification information. For integrity-preservation frameworks, cryptographic hash values, cyclic permutation mappings, inter-block verification dependencies, and cross-frame verification relationships are generated. The resulting information is subsequently

forwarded to the embedding module.

- **Security Information Embedding:** The generated security information is embedded into the visual or auditory components of the video object according to the selected methodology. The resulting watermarked video can subsequently be stored, transmitted, and verified while preserving its security properties.
- **Security Information Extraction, Verification, and Decision Making:** The embedded information is recovered from the watermarked video object and utilized to determine its authenticity or integrity status. For authenticity-preservation models, the embedded watermark information and frame-to-audio verification information are extracted and validated against the expected values. For integrity-preservation frameworks, the embedded cryptographic verification information is recovered and compared against the recomputed hash values of the corresponding frame blocks. Any inconsistency indicates potential tampering of the associated video regions. Furthermore, the established cross-frame verification dependencies enable the detection of temporal manipulations, including frame insertion, frame deletion, and frame swapping attacks. When applicable, the framework additionally localizes the affected frame blocks and produces the final authenticity or integrity decision.

## 5.5 Document Authenticity and Integrity

The proposed document authenticity-preservation framework exploits the structural and visual characteristics of PDF documents in order to embed and verify authenticity information while preserving document readability and functionality. More specifically, the framework operates on document pages and textual elements, utilizing dynamically generated QR codes together with scrambling and permutation mechanisms to construct authenticity information. The common processing pipeline consists of four stages, namely document segmentation, information generation, information embedding, and information extraction, verification, and decision making. Together, these stages provide the mechanisms required for document authenticity verification while maintaining the visual appearance and structural integrity of the protected document.

- **Document Segmentation:** The input PDF document is decomposed into its constituent pages and textual elements. These document components constitute the processing units utilized throughout the authenticity-preservation procedure and are subsequently forwarded to the security information generation module.
- **Security Information Generation:** Using the extracted document pages, the framework generates the authenticity information required for document protection. More specifically, the selected authenticity information is first encoded into a QR code. Subsequently, the generated QR code is resized according to the embedding requirements of the document and transformed through Arnold scrambling and permutation procedures in order to increase security and conceal its visual structure. The resulting transformed QR-code representation constitutes the watermark information that will be embedded into the document and is therefore forwarded to the embedding module.
- **Security Information Embedding:** The generated QR-code representation is embedded into the PDF document through controlled modifications of selected textual elements. More specifically, the embedding procedure alters the color values of selected document characters according to the values of the transformed QR-code representation while preserving document readability, functionality, and visual quality. The resulting output corresponds to the final watermarked PDF document.
- **Information Extraction, Verification, and Decision Making:** The embedded authenticity information is recovered from the examined PDF document by analyzing the modified textual elements and reconstructing the corresponding QR-code representation. Subsequently, the inverse permutation and Arnold transformation procedures are applied in order to recover the original authenticity information. The recovered information is then validated against the expected authenticity data in order to determine the authenticity status of the examined document. Based on the verification outcome, the framework produces the final authenticity decision.

# CHAPTER 6

## EVALUATION

- 
- 6.1 Data-set Description
  - 6.2 Quality, Robustness and Fragility Metrics
  - 6.3 Computational Efficiency Metrics
  - 6.4 Attack Vector
  - 6.5 Experimental Results on Images
  - 6.6 Experimental Results on Audio
  - 6.7 Experimental Results on Video
  - 6.8 Experimental Results on Documents
- 

### 6.1 Data-set Description

For the evaluation of the authenticity and integrity preservation capabilities of the proposed watermarking systems, different data-sets were utilized depending on the type of digital object under examination.

**Image Data-sets:** In the case of digital images, the image data-set was divided into three categories: a general-purpose image data-set, a medical image data-set, and a forensic image data-set. Furthermore, the image data-sets were divided into multiple resolution categories, ranging from  $512 \times 512$  images up to native 4K resolution, in

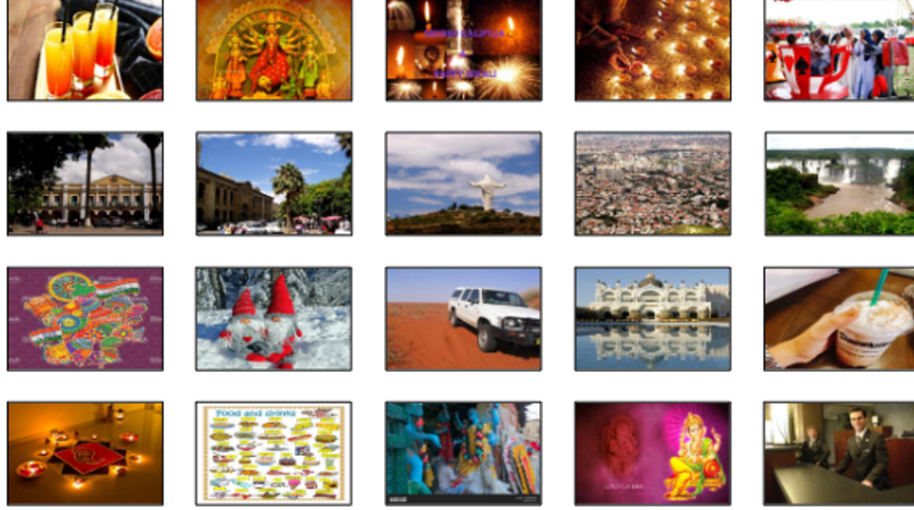


Figure 6.1: Example images from the general-purpose data-set utilized in the evaluation process, consisting of multiple image types.

order to better assess the proposed watermarking algorithms and reduce resolution-dependent bias in the evaluation process. Bellow we provide more details about the different image data-sets.

- **General purpose image data-set:** For the evaluation of the authenticity and integrity preservation capabilities of the proposed watermarking systems on general-purpose images, we utilized the data-set defined in [64]. The data-set consists of four image categories: Art & Culture, Architecture, Food and Drinks, and Travel and Adventure. All images were stored in RGB color space using both .jpeg and .png format. In Figure 6.1 we present an example of the generic images used in our evaluation.
- **Medical image data-set:** For the evaluation of the authenticity and integrity of medical images we utilized different data-sets including different medical image modalities. More specifically we used X-ray images defined in [65], CT-scan images defined in [66], MRI images defined in [67], Ocular/Retinal images defined in [68] and Endoscopy images defined in [69]. All images were stored in RGB color space using both .jpeg and .png format. In Figure 6.2 we present an example of the medical images used in our evaluation.
- **Forensic image data-set:** For the evaluation of the authenticity and integrity preservation of forensic images, Pexels was utilized as the primary source for

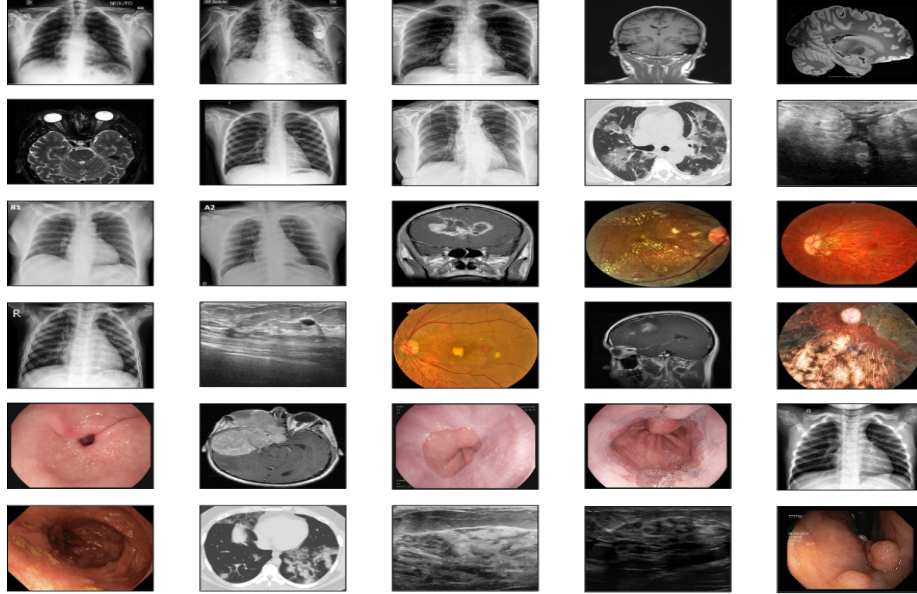


Figure 6.2: Example images from the medical image data-set utilized in the evaluation process, consisting of multiple image modalities.



Figure 6.3: Example images from the forensic image data-set utilized in the evaluation process, consisting of multiple forensic scenes.

Table 6.1: Characteristics of the VGG Sound video data-set

| Resolution        | Frame Rates (fps) | Codecs           | Duration (seconds) |
|-------------------|-------------------|------------------|--------------------|
| 720p (1280×720)   | 24, 30, 60        | H.264, HEVC, AV1 | 10                 |
| 1080p (1920×1080) | 24, 30, 60        | H.264, HEVC, AV1 | 10                 |
| 2K (2560×1440)    | 24, 30, 60        | H.264, HEVC, AV1 | 10                 |
| 4K (3840×2160)    | 24, 30, 60        | H.264, HEVC, AV1 | 10                 |

the selection of the forensic image dataset [70]. The selected images contained significant visual clues related to forensic scenes and investigations. All images were stored in RGB color space using the .png format. In Figure 6.3 we present an example of the forensic images used in our evaluation.

**Audio Data-sets:** For the evaluation of the proposed authenticity and integrity preservation models for audio signals, a diverse collection of audio tracks freely available on the web was utilized. More specifically, the data-set included music-only recordings, music accompanied by vocals, and speech-only audio tracks, with durations ranging from 30 seconds to 1 minute. Furthermore, audio recordings obtained from publicly available broadcasts on YouTube were also considered in order to assess the effectiveness of the proposed methodologies on content of higher practical value and increased susceptibility to malicious manipulation. The utilization of different audio categories allows a more comprehensive evaluation of the proposed models under realistic operating conditions and varying acoustic characteristics.

**Video Data-sets:** To evaluate the proposed watermarking techniques for the authenticity and integrity preservation of digital videos, two different datasets were utilized, namely the VGGSound dataset [71] and the CCTV dataset [72]. The VGGSound dataset consists of general-purpose videos collected from the YouTube platform, covering a wide range of real-world scenarios and audio-visual content. In contrast, the CCTV dataset contains surveillance videos capturing criminal activities of different categories, such as assault, shooting, and arson incidents. For our evaluation using the VGG Sound data-set, we split the videos into 4 resolution groups with different frame rates and codecs. For our evaluation using the CCTV data-set we selected 10 videos with different durations. In Table 6.1 and Table 6.2 we present the characteristics of

Table 6.2: Characteristics of the CCTV video data-set

| Video  | Resolution       | Codec | Frame Rate(fps) | Duration(s) |
|--------|------------------|-------|-----------------|-------------|
| cctv1  | $320 \times 240$ | H264  | 30              | 8           |
| cctv2  | $320 \times 240$ | H264  | 30              | 59          |
| cctv3  | $320 \times 240$ | H264  | 30              | 35          |
| cctv4  | $320 \times 240$ | H264  | 30              | 61.8        |
| cctv5  | $320 \times 240$ | H264  | 30              | 45          |
| cctv6  | $320 \times 240$ | H264  | 30              | 40          |
| cctv7  | $320 \times 240$ | H264  | 30              | 25          |
| cctv8  | $320 \times 240$ | H264  | 30              | 30          |
| cctv9  | $320 \times 240$ | H264  | 30              | 44          |
| cctv10 | $320 \times 240$ | H264  | 30              | 26          |

the two data-sets.

**PDF Data-sets:** For the evaluation of the proposed PDF authenticity-preservation model, a diverse collection of PDF documents was utilized. More specifically, the data-set consisted of publicly available PDF files obtained from online repositories, documents generated manually for experimental purposes, and synthetic PDF documents created using Artificial Intelligence tools. The generated and collected documents included different layouts, font styles, text densities, and graphical elements in order to evaluate the effectiveness of the proposed methodology under a wide range of document characteristics. This diverse data-set enabled the assessment of the proposed framework in realistic document-processing scenarios while ensuring the reproducibility and flexibility of the experimental evaluation.

## 6.2 Quality, Robustness and Fragility Metrics

The evaluation of watermarking systems requires the utilization of multiple metrics for assessing the quality of the watermarked digital object, the robustness of the watermarking scheme against different malicious attacks, and the fragility of the system in order to detect unauthorized modifications and tampering attempts. More specifically, quality and imperceptibility assessment is performed between the original



and the watermarked digital object in order to evaluate the degradation introduced by the embedding procedure. Robustness is evaluated following the simulation of malicious attacks on the watermarked digital object, assessing the capability of the watermarking system to successfully extract the embedded watermark with minimal degradation. Finally, fragility metrics are computed after the simulation of tampering attacks in order to evaluate the accuracy of the watermarking system in detecting and precisely localizing modified or manipulated regions of the watermarked digital object. In this work, in order to evaluate the proposed watermarking techniques for the preservation of authenticity and integrity of digital objects, we employ a set of well-established quantitative metrics that assess both the perceptual quality of the protected objects and the effectiveness of the authenticity and integrity verification procedures. More specifically, perceptual quality and watermark robustness are evaluated through the Peak Signal-to-Noise Ratio (PSNR), the Structural Similarity Index Measure (SSIM), the Normalized Correlation (NC), the Bit Error Rate (BER), and the Extraction Rate ( $R_e$ ). Furthermore, for the evaluation of the proposed integrity-preservation and tamper-detection frameworks, we utilize the classification metrics Precision, Recall, F1-score, and Accuracy, which quantify the effectiveness of the proposed methodologies in identifying and localizing malicious modifications. Together, these metrics provide a comprehensive assessment of imperceptibility, robustness, extraction reliability, tamper-detection performance, and overall verification effectiveness.

### 6.2.1 Quality and Imperceptibility Measurement

To evaluate the imperceptibility and perceptual quality of the watermarked digital objects, the PSNR, SSIM and NC metrics were utilized. The PSNR and SSIM metric was employed for the evaluation of digital images and digital videos, while the NC along with PSNR and SSIM metrics was utilized for the evaluation of QR codes embedded within PDF documents. Next, the formal definitions of the above metrics are presented.

The PSNR metric, measured in decibels ( $dB$ ), is utilized to evaluate the distortion between the original digital object  $DO_o$  and the watermarked digital object  $DO_w$ . The PSNR metric is defined as follows:

$$\text{PSNR} = 20 \cdot \log_{10} \left( \frac{255}{\sqrt{\text{MSE}}} \right). \quad (6.1)$$

PSNR calculates the ratio of the peak signal power to the Mean Squared Error (MSE) between the original and watermarked image/video frame, where the MSE is given by:

$$\text{MSE} = \frac{1}{N \cdot M} \sum_{n=1}^N \sum_{m=1}^M |DO_o(n, m) - DO_w(n, m)|^2. \quad (6.2)$$

PSNR values above 30 dB are generally considered indicative of good perceptual quality for the watermarked digital object, suggesting that the embedding process introduces relatively low distortion.

The SSIM metric evaluates the structural similarity between the original and the watermarked digital object by utilizing the factors of Luminance ( $L$ ), Contrast ( $C$ ), and Structure ( $S$ ). For a pair of digital objects, denoted as  $DO_o$  and  $DO_w$ , corresponding to the original and the watermarked object respectively, the SSIM metric is computed as follows:

$$\text{SSIM}(DO_o, DO_w) = L(DO_o, DO_w) \cdot C(DO_o, DO_w) \cdot S(DO_o, DO_w), \quad (6.3)$$

SSIM values range between  $-1$  and  $1$ , with values close to  $1$  indicating near-perfect structural similarity and perceptual quality between the original and the watermarked digital objects. In contrast, values close to  $0$  or below typically indicate significant degradation introduced during the embedding process.

The NC metric evaluates the similarity between two digital objects by measuring the correlation between their corresponding values. In the context of watermarking systems, NC is commonly utilized to assess the similarity between the original and the extracted watermark, or in our QR code embedding, the correlation between the original and the reconstructed QR code. For a pair of watermarks, denoted as  $W_{em}$  and  $W_{ex}$ , corresponding to the original and the extracted watermark respectively, the NC metric is computed as follows:

$$\text{NC} = \frac{\sum_i W_{em_i} W_{ex_i}}{\sqrt{\sum_i W_{em_i}^2 \sum_i W_{ex_i}^2}}. \quad (6.4)$$

NC values range between  $-1$  and  $1$ . Higher NCC values indicate stronger similarity and more accurate watermark extraction, while lower values indicate increased distortion or degradation of the extracted information.

### 6.2.2 Robustness Measurement

To assess the robustness performance of our proposed watermarking algorithms against a wide range of attack scenarios, we utilized two robustness metrics namely the Extraction Rate ( $R_e$ ) and the Bit Error Rate (BER). The  $R_e$  metric was utilized for the evaluation of authenticity preservation in digital images and digital videos in cases where a sequence of integers encoded as SiPs was embedded into the host digital object. On the other hand, the BER metric was employed for the evaluation of the extracted SiPs, as well as for the evaluation of QR codes extracted from PDF documents. More specifically, two variations of the BER metric were defined in this research study: the first variation evaluates the BER of the extracted SiPs, while the second evaluates the BER of the extracted QR codes. Next, the formal definitions of robustness metrics are presented.

The Extraction Rate ( $R_e$ ) metric is utilized to evaluate the accuracy of the extracted integer sequence in comparison with the originally embedded sequence. Let  $S_a$  denote the original embedded integer sequence and  $S_b$  the extracted integer sequence after the watermark extraction process. The  $R_e$  metric is computed as follows:

$$R_e = \frac{1}{|S_a|} \sum_{i=0}^{|S_a|-1} \mathbb{I}(S_a(i) = S_b(i)), \quad (6.5)$$

The  $R_e$  metric provides a direct indication of the authenticity of the digital object (image or video frame), with  $R_e = 100\%$  indicating perfect extraction, while lower values suggest potential tampering or degradation due to attacks.

The BER metric is utilized to quantify the proportion of incorrectly extracted bits with respect to the total number of embedded bits. In the proposed frameworks, BER is computed either between the original embedded integer sequence and the corresponding extracted sequence, or between the original QR-code representation and the recovered QR code, depending on the employed watermarking methodology. Lower BER values indicate higher extraction accuracy and stronger robustness against distortions and malicious modifications.

**Bit Error Rate based on SiP.** Let an integer embedded in the form of SiP of size  $P$ . Each element of the SiP is represented in its binary form using a fixed number of bits. By concatenating the binary representations of all elements of the SiP, a bit-string  $b_{sip}$  is constructed. For example, if we want to embed the integer  $W = 10$  whose SiP is  $[5, 7, 9, 8, 1, 6, 2, 4, 3]$ , then the corresponding binary representation of

each integer in the sequence is  $5 = 0101$ ,  $7 = 0111$ , ...,  $3 = 0011$ . Consequently, the bit string will be equal to  $b_{sip} = 01010111...0011$ , which is the binary representation of  $[5, 7, 9, 8, 1, 6, 2, 4, 3]$ . Finally, let  $b_{sip}$  and  $\hat{b}_{sip}$  denote the embedded and extracted bit-strings, respectively, an error counter  $EC$  is utilized to accumulate the number of mismatched bits between  $b_{sip}$  and  $\hat{b}_{sip}$ . Given a sequence of integers represented as SiPs having length  $D$ , the BER is computed as:

$$\text{BER}_{\text{sip}} = \frac{C_{er}}{|b_{sip}| \cdot D} \quad (6.6)$$

**Bit Error Rate based on pixel values.** Let a binary QR-code representation of dimensions  $M \times M$ , consisting of a total of  $T = M^2$  pixels, be embedded within a PDF document. After the extraction procedure, let  $\hat{Q}$  denote the recovered QR-code representation of the same dimensions. In order to quantify the pixel mismatches between the original and the extracted QR codes, an error counter  $EC$  is defined, which accumulates the number of mismatched pixels between the two QR codes. The BER can be computed as:

$$\text{BER}_{\text{bin}} = \frac{EC}{T}, \quad (6.7)$$

BER values range between 0 and 1. Values close to 0 indicate highly accurate and robust watermark extraction, while higher BER values indicate increased degradation of the extracted watermark. In particular, BER values above 0.3 typically indicate significant watermark degradation and consequently reduced authenticity preservation capability.

### 6.2.3 Fragility Measurement

To evaluate the fragility of the proposed watermarking systems for digital object integrity preservation, standard classification-based metrics were utilized. These metrics effectively assess the capability of the proposed schemes in detecting and localizing tampered regions. More specifically, the evaluation process is based on the Precision, Recall, F1-score, and Accuracy metrics, which are computed by comparing the tampered regions against the corresponding ground-truth masks at block level/video frame level. Let TP (True Positives) denote the number of correctly identified manipulated blocks, FP (False Positives) the number of incorrectly detected manipulated blocks, FN (False Negatives) the number of undetected manipulated blocks, and TN

(True Negatives) the number of correctly identified authentic blocks. In case of only tamper detection and note localization the above metrics correspond to frame level manipulations.

The Precision metric evaluates the proportion of correctly detected tampered blocks/video frames among all detected tampered blocks/video frames, and is defined as:

$$\text{Precision} = \frac{TP}{TP + FP}. \quad (6.8)$$

Recall measures the proportion of actual tampered blocks/video frames that are correctly detected:

$$\text{Recall} = \frac{TP}{TP + FN}. \quad (6.9)$$

The F1-score provides a balanced measure between Precision and Recall, and is defined as:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (6.10)$$

Intersection over Union (IoU) measures the overlap between the tampered blocks/video frames and the ground truth:

$$\text{IoU} = \frac{TP}{TP + FP + FN} \quad (6.11)$$

Finally, Accuracy measures the overall proportion of correctly classified blocks/video frames (both tampered and untampered), and is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \quad (6.12)$$

Precision values close to 1 indicate that the majority of the detected tampered regions correspond to actual manipulations, minimizing false detections. Similarly, Recall values close to 1 demonstrate that the detection process successfully identifies most of the manipulated regions with minimal missed tampering areas. Accuracy evaluates the overall reliability of the watermarking system in correctly classifying both authentic and manipulated regions. Finally, the F1-score provides a balanced evaluation between Precision and Recall, indicating the capability of the watermarking system to effectively detect tampered regions while simultaneously minimizing false detections and missed manipulations.

### 6.3 Computational Efficiency Metrics

An equally important aspect of the evaluation of the proposed work is computational efficiency. Although a watermarking system may effectively preserve the authenticity and integrity of digital objects, it must also remain computationally efficient and scalable in order to be applicable in real-world scenarios and large-scale digital environments. For the evaluation of computational efficiency, standard benchmarking metrics were utilized in order to assess the embedding and extraction execution times, as well as the computational resource requirements of the proposed watermarking systems, including CPU and RAM utilization.

**Embedding and Extraction Time.** The embedding and extraction execution times of the proposed watermarking systems were calculated by measuring the total time required for the main embedding and extraction procedures to complete their execution. In cases where both authenticity and integrity watermarking mechanisms were applied simultaneously, the corresponding embedding and extraction procedures were evaluated independently, and their execution times were subsequently combined in order to determine the total embedding and extraction time of the watermarking system. In order to obtain more reliable and consistent computational efficiency measurements, the embedding and extraction procedures were executed multiple times, and the final reported execution times correspond to the average values across all iterations.

**Computational Resources.** Computational resources constitute a critical factor in the deployment of watermarking systems within real-world applications. Resource requirements such as CPU utilization and memory consumption should remain as low as possible in order to ensure the scalability and efficient deployment of the watermarking system in complex computational environments. In this direction, in order to assess the computational resource consumption of the intensive procedures of the proposed watermarking systems, CPU utilization and RAM memory consumption were monitored during the execution of computationally demanding functions involving repetitive transformations, such as multiple iterations of the Arnold transformation. To obtain more reliable and consistent measurements, the monitoring process was repeated multiple times and the final reported values correspond to the average resource utilization across all executions.

## 6.4 Attack Vector

The evaluation framework of the proposed watermarking techniques is organized based on the two primary functionalities of the system, namely robust authenticity preservation and fragile integrity preservation. Consequently, the applied attacks are divided into two major categories: Authenticity Attacks and Integrity Attacks. Authenticity Attacks are utilized to attest the robustness of the embedded watermark against common signal-processing operations and malicious manipulations, whereas Integrity Attacks are employed to evaluate the sensitivity of the embedded integrity information to unauthorized modifications, as well as the capability of the system to accurately detect and localize tampered regions. The following subsections present the different attack categories and the corresponding distortions applied for the comprehensive evaluation of the proposed watermarking techniques under diverse attack scenarios.

### 6.4.1 Filtering Attacks

Filter-based attacks simulate common image and video-frame processing operations that may affect the robustness of the embedded authenticity information. In our evaluation, representative filtering operations were considered, including Gaussian Blur (GB), Median Blur (MB), and Average Blur (AB). In addition, noise and intensity transformation attacks were applied, namely Gaussian Noise (GN), Salt & Pepper Noise (SP), Gamma Correction (GC), and Histogram Equalization (HEQ). Collectively, these attacks cover a wide range of degradation scenarios, including smoothing, noise addition, contrast enhancement, and intensity transformations, enabling a comprehensive evaluation of watermark robustness under realistic signal-processing conditions.

Similarly, for the evaluation of the proposed audio watermarking models, representative audio signal-processing attacks were deployed, including amplification (*Amplify*), normalization (*Normalize*), and fade-in/fade-out filtering operations. These attacks simulate common audio enhancement and post-processing procedures that modify the amplitude characteristics of the signal and may affect the embedded authenticity information. Together, the selected image, video, and audio filtering attacks provide a comprehensive assessment of the robustness of the proposed watermarking frameworks against realistic multimedia processing operations.

### 6.4.2 Compression Attacks

Compression attacks aim to reduce the size of the digital object while potentially degrading the embedded watermark information. In the conducted experiments, JPEG compression was applied to digital images and video frames using quality factors of  $Q = 50\%$ ,  $70\%$ , and  $90\%$  in order to evaluate the robustness of the embedded authenticity information under different compression levels. For digital video objects, MPEG-4 and H.264 compression attacks were utilized with varying compression rates. Additionally, re-encoding operations were applied to the watermarked video objects in order to evaluate the robustness of the embedded watermark under codec-based compression and video transcoding operations. In the case of audio signals, audio-specific compression operations were applied through down-sampling using different sampling rates as also re-sampling to different sampling rates (linear and non-linear). For PDF files we used pdf compression with low, medium and high quality factors.

### 6.4.3 Geometric and Document-processing Attacks

Geometric attacks involve spatial transformations that modify the structure of the digital object and may disrupt the alignment and extraction of the embedded authenticity information. Cropping attacks were applied in both the horizontal and vertical directions. More specifically, left, intermediate, and right region cropping operations were performed using cropping percentages of  $25\%$ ,  $50\%$ , and  $75\%$  of the original image dimensions in order to evaluate the robustness of the embedded authenticity information under severe geometric distortions. In the case of video objects, the robustness of the proposed watermarking model was additionally evaluated under upscaling and downscaling attacks. Starting from a standard video resolution of  $1080p$ , the videos were upscaled to  $1440p$  and downscaled to  $720p$  in order to assess the robustness of the embedded authenticity information under resolution transformation operations. For PDF documents, geometric and document-layout transformations were additionally considered. More specifically, page rotation operations, page scaling, page resizing, and margin-modification attacks were applied in order to evaluate the robustness of the embedded QR-code watermark under document-structure transformations. Furthermore, page cropping operations were performed on the PDF pages utilizing different cropping percentages in both the horizontal and vertical directions. Such



attacks may alter the spatial arrangement of the embedded textual information and potentially affect the extraction and reconstruction of the embedded QR-code representation.

#### **6.4.4 Content Modification and Tampering Attacks**

For the evaluation of image integrity, insertion and removal attacks were applied. In insertion attacks, external image objects were inserted into the watermarked image, whereas in removal attacks specific spatial regions were removed from the watermarked image in order to evaluate the capability of the proposed watermarking system to detect and localize tampered areas. Similarly, for PDF documents, content-modification attacks were performed through text insertion, text removal, and text-modification operations at different positions of the document pages, including the beginning, middle, and end of the textual content. Furthermore, OCR-free image conversion attacks were additionally considered, where the PDF document was transformed into image format without preserving the embedded textual structure. Such attacks directly affect the textual regions responsible for the QR-code embedding process and therefore evaluate the capability of the proposed framework to detect malicious modifications and authenticity violations within the protected PDF document.

#### **6.4.5 Temporal Attacks**

Temporal attacks include operations intended to disrupt the temporal continuity and consistency of sequential digital data, particularly in digital video and audio objects. In the evaluation of the proposed watermarking model for digital video objects integrity preservation, Frame Swap (FS), Frame Drop (FD), Frame Rate Increase (FRI) and Frame Rate Decrease (FRD) attacks were employed using attack percentages of 10%, 20%, and 30%. In Frame Swap attacks, randomly selected video frames were exchanged with one another, whereas in Frame Drop attacks randomly selected frames were removed from the video sequence. In Frame Rate Increase and Frame Rate Decrease attacks, starting from a video with a standard frame rate of 30fps we increased the frame rate to 60fps and also decrease it to 24fps.

### 6.4.6 AI-based Attacks

In recent years, the rapid development of Artificial Intelligence (AI)-based image generation and editing frameworks has significantly increased the realism and accessibility of multimedia manipulations. Consequently, AI-driven attacks constitute an important category of adversarial modifications that should be considered during the evaluation of authenticity-preservation and integrity-verification systems. In this work, we evaluate the robustness and tamper-detection capabilities of the proposed frameworks against a series of AI-assisted manipulation attacks involving object removal, object insertion, content generation, and semantic image editing. More specifically, AI-based inpainting techniques were utilized in order to remove objects or regions from multimedia content while reconstructing visually plausible background information. In particular, frameworks such as *LaMa* were deployed for semantic inpainting and object removal, enabling realistic restoration of manipulated regions through deep-learning-based content synthesis. Furthermore, generative models based on diffusion architectures, such as *Stable Diffusion*, were utilized to generate and insert synthetic objects or image regions into the protected multimedia content, simulating advanced forgery and content-generation attacks.

Additionally, conventional intelligent editing tools were also considered, including selection-based object manipulation and copy-paste operations utilizing software-assisted segmentation mechanisms such as the *Magic Select* functionality available in image-editing environments. Such approaches enable semi-automatic extraction, relocation, insertion, or removal of visual objects, facilitating localized manipulations without requiring advanced manual editing skills. The deployment of AI-assisted attacks is particularly important in authenticity and integrity-preservation evaluation scenarios, since modern generative and inpainting frameworks are capable of producing visually coherent modifications that are difficult to detect through perceptual inspection alone. Consequently, the proposed attacks simulate realistic adversarial scenarios where malicious users attempt to modify multimedia content while preserving its apparent visual consistency.

### 6.4.7 Combined Attacks

In the final category of attacks, multiple attack operations were combined in order to evaluate the robustness and stability of the proposed watermarking models un-

der multi-attack scenarios. More specifically, sequences of different signal-processing, compression, filtering, geometric, and temporal attacks were applied consecutively to the same digital object, simulating realistic conditions where digital content may undergo multiple modifications stages. Specifically in the case of images we combined insertion and removal attacks were in the same watermarked image we inserted an external object and then we removed a specific image region. In the case of video objects, combined attacks may involve successive modifications of the video's resolution, frame rate, and codec. More specifically, a video may first be upscaled or downscaled, then converted to a different frame rate, and finally re-encoded using another codec. Such transformation chains frequently occur when videos are uploaded to social media platforms, where automated processing mechanisms modify the original video to meet platform-specific requirements.

## **6.5 Experimental Results on Images**

In this section, the experimental results regarding the proposed image-based authenticity-preservation and integrity-preservation frameworks are presented and analyzed. More specifically, the evaluation focuses on the computational efficiency, perceptual quality, authenticity verification capability, and tamper detection and localization performance of the proposed watermarking models under different attack categories and signal-processing conditions.

The following image-based models are experimentally evaluated:

- **Repetitive SiP Watermarking Model**
- **Code-based Multiple SiP Watermarking Model**
- **ROMFAIA Hybrid Watermarking Model**
- **ProtAIN Dual NFT watermarking Model**
- **Cross-block Permutation-based Watermarking Model**

### **Repetitive SiP Watermarking Model - Methodology**

In the experimental setup followed for the evaluation of the proposed repetitive watermarking technique we distinguish four categories of experiments regarding the

crop attack. We tested the crop cases where the crop has been performed vertically, horizontally, either in a side of the image or by cropping an intermediate region taking into account various percentages of crop. Moreover we compare the PSNR and SSIM values exhibited on the images when they have been watermarked utilizing the SiP technique as proposed in [1], and the repetitive one proposed in this work, after a series of compression attacks. We also computed the embedding time in comparison with the base approach presented in [1]. The data-set (see Figure 6.4) utilized for the evaluation of our proposed model contains images in jpeg format of dimensions  $4096 \times 4096$  and  $8192 \times 8192$ . The proposed repetitive watermarking technique is developed in Python 3.10 using the Jupyter framework, and the corresponding evaluation experiments were conducted in a commodity computer with 9<sup>th</sup> Gen. Intel i3-9500 (6 cores - 6 threads) and 8 GB of RAM running over Windows 10.

### **Code-based Multiple SiP Watermarking Model - Methodology**

For the experimental design followed for the evaluation of our code-based multiple watermarking model we consider a wide range attack vector including various attacks such as Salt and Pepper (S&P), Gaussian Blur (G\_BL), Histogram Equalization (HEQ), Compression (CMP). Through the evaluation of our model for the deployment of each type of attack we investigated (where applicable) various factors related to the effectiveness of each type of attack, i.e., for the case of Gaussian Blur using  $3 \times 3$  and  $5 \times 5$  kernel or for the case of Compression using Quality=75, 65, and 55. The data-set of forensic digital images used for the evaluation of the proposed technique contained 5 forensic images as they are illustrated in Figure 6.3 (4K rgb-images in .jpg format) including significant clues related to forensic scenes. We utilized a  $6 \times 6$  *plane* to partition the image into 36 *authenticity-blocks* and a  $9 \times 9$  *grid* corresponding to integers in the range [8-15], to further partition each block to *cells*. The proposed watermarking technique was developed in Python 3.10 using the Jupyter framework, and the corresponding evaluation experiments were conducted in a commodity computer with 9<sup>th</sup> Gen. Intel i3-9100 (4 cores - 4 threads) and 12 GB of RAM running over Windows 10.

### **ROMFAIA Hybrid Watermarking Model - Methodology**

The experimental evaluation was designed to assess the effectiveness of the proposed framework in preserving both the authenticity and integrity of medical images. Since the framework consists of two complementary security layers, namely authenticity

preservation and integrity preservation, the evaluation methodology was structured according to these two components.

For authenticity preservation, each image was partitioned into a  $4 \times 4$  *integrity-plane*, resulting in 16 non-overlapping *authenticity-block*. A code-sequence consisting of 16 integers in the range  $[8, 15]$  was generated and embedded into the image, with one integer assigned to each *authenticity-block*. Each integer was represented by a SiP of size 9, corresponding to a  $9 \times 9$  embedding grid. The same code-sequence was employed throughout the evaluation in order to ensure consistent comparisons among different images and attack scenarios.

For integrity preservation, the resulting watermarked image was partitioned into an adaptive *integrity-plane*. More specifically, for an image of dimensions  $M \times N$ , the greatest common divisor  $g = \gcd(M, N)$  was utilized to determine the maximum valid partition size satisfying the constraint that each *integrity-block* contained at least 256 pixels. This requirement guarantees sufficient capacity for embedding the 256-bit output of the  $HMAC-SHA_{256}$  function. For each *integrity-block*, a cryptographic digest was generated from the red channel and embedded into the least significant bits of the green channel of another *integrity-block* according to a randomly generated permutation. The resulting permutation established a set of block-to-block verification relationships and the corresponding  $\Lambda$ -relations utilized during tamper detection and localization.

The authenticity component was evaluated under a variety of signal-processing and geometric attacks. More specifically, crop attacks were performed at severity levels of 25%, 50%, and 75% along different image directions. Filtering attacks included Gaussian Blur, Median Blur, Mean Filtering, Sharpening, Histogram Equalization, Gamma Correction, and Gaussian Noise with variances 0.05, 0.1, and 0.2. Compression robustness was evaluated through JPEG compression with quality factors  $Q = 90\%$ ,  $70\%$ , and  $50\%$ . Furthermore, geometric attacks including image scaling and translation were applied in order to evaluate the robustness of the embedded authenticity information.

The integrity component was evaluated through deliberate tampering operations designed to alter the semantic content of the medical images. These attacks included object insertion, object removal, and combined insertion-removal modifications. Following each manipulation, the extracted cryptographic verification information and the generated  $\Lambda$ -relations were analyzed in order to identify inconsistencies among

the corresponding *Integrity-blocks* and localize the manipulated regions.

The evaluation dataset (see 6.2) consisted of medical images belonging to multiple imaging modalities, including brain MRI scans, chest X-ray images, CT scans, gastrointestinal endoscopy images, and retinal fundus images. The images covered a wide range of resolutions, including 720p, 1024p, 1080p, 2K, 4K, and randomized dimensions, allowing the proposed framework to be evaluated under diverse spatial configurations. Finally, image quality, robustness, tamper localization accuracy, and computational performance were assessed through standard perceptual, security, and execution-time metrics. All experiments were conducted on a system equipped with an AMD Ryzen 7 5700G CPU (8 cores, 16 threads), 32 GB RAM, running Windows 11 Pro.

### **ProtAIN Dual NFT watermarking Model - Methodology**

The deployment of the proposed ProtAIN framework follows a three-phase methodology that first secures the digital image before establishing its presence on the blockchain. Initially, the original image  $I$  undergoes authenticity preservation through the frequency-domain embedding of the 16-digit user code, producing the authenticity-watermarked image  $I_w$ . Subsequently, fragile integrity information is embedded into  $I_w$  in the spatial domain following the ROMFAIA framework, resulting in the final protected image  $I_{wt}$ , which simultaneously provides authenticity and integrity guarantees.

Once the image has been secured, the framework proceeds with the deployment of a backup NFT. More specifically, a dual image  $I' * wt$  is generated by applying the Arnold transformation  $K$  consecutive times to  $I * wt$ , where  $K$  is a secret transformation key. The transformed image is uploaded to IPFS, while its corresponding smart contract is deployed on the blockchain, establishing a secure and independently verifiable backup prior to the public release of the NFT.

Finally, the original NFT is deployed by uploading the protected image  $I_{wt}$  to IPFS and deploying its corresponding smart contract on the blockchain before making it publicly available through the NFT marketplace. By securing the image, establishing a verifiable backup, and subsequently releasing the original NFT, the proposed methodology ensures that authenticity, integrity, and an independent verification reference are available before the digital asset becomes publicly accessible.

### **Cross-block Permutation-based Watermarking Model - Methodology**

The data-set utilized for the evaluation of the proposed tamper detection and localization model consisted of 6 forensic digital images, as illustrated in Figure 6.3, re-sized to  $512 \times 512$  pixels. The images were pre-processed and stored in .png format, since lossless image representations are required in integrity-preservation and fragile watermarking frameworks in order to avoid unintended compression affecting the embedded integrity information. Each image was partitioned into 1024 non-overlapping *integrity-blocks* using a  $32 \times 32$  grid, resulting in *integrity-blocks* containing 256 pixels, which provide sufficient capacity for embedding the generated 256-bit hash values into their corresponding LSB regions. Furthermore, the inter-block verification dependencies were established through the deployment of random permutations of cardinality 1024, enabling the formation of distributed verification relations among the corresponding *integrity-blocks*.

Throughout the experimental evaluation, emphasis was placed both on the capability of the proposed framework to fully detect whether a forensic image had been tampered and on the precision of the localization procedure, ensuring that the tampered regions were accurately identified without introducing false-positive detections. More specifically, three main attack categories were considered during the evaluation process. In *Insertion Attacks*, external forensic objects or image regions were inserted into the image in order to simulate malicious content addition. In *Removal Attacks*, existing forensic elements or image regions were removed from the image, simulating concealment or deletion attempts. Finally, *Combined Attacks* simultaneously incorporated both insertion and removal operations within the same forensic image, either affecting different image regions or overlapping spatial areas, enabling the evaluation of the proposed framework under more complex tampering scenarios. The proposed watermarking technique was developed in Python 3.10 using the Jupyter framework, and the corresponding evaluation experiments were conducted in a commodity computer with 9<sup>th</sup> Gen. Intel i3-9100 (4 cores - 4 threads) and 12 GB of RAM running over Windows 10.

### 6.5.1 Images: Run-time Efficiency

#### Repetitive SiP Watermarking Model

Table 6.3 presents the embedding computational time of the proposed repetitive SiP watermarking model for images of different resolutions. The results demonstrate that

Table 6.3: Embedding computational time of the proposed repetitive SiP watermarking model.

| Image      | Image Size         | Repetitive SiP Time (s) |
|------------|--------------------|-------------------------|
| people     | $4096 \times 4096$ | 84.14                   |
| horizon    | $4096 \times 4096$ | 83.87                   |
| lighthouse | $8192 \times 8192$ | 513.70                  |
| elephant   | $8192 \times 8192$ | 504.50                  |

the proposed approach remains computationally efficient even when processing high-resolution images. For images of size  $4096 \times 4096$ , the embedding procedure requires approximately 84 seconds, while for ultra-high-resolution images of size  $8192 \times 8192$ , the execution time ranges from approximately 500 to 514 seconds.

The observed computational performance is mainly attributed to the hierarchical partitioning strategy employed by the proposed model. More specifically, in addition to the *plane* partitioning, an additional *grid* structure is introduced within each *cell*, resulting in substantially smaller processing regions. Consequently, the FFT-based embedding operations are performed over smaller spatial areas, reducing the computational cost of the frequency-domain transformations. For example, for an image of size  $8192 \times 8192$  and a *plane* of size  $7 \times 7$ , the additional  $7 \times 7$  *grid* partitioning reduces the effective processing regions to approximately  $167 \times 167$  pixels. This reduction significantly decreases the computational burden of the embedding procedure, enabling the proposed model to efficiently process very high-resolution images while maintaining its watermarking capabilities.

### Code-based Multiple SiP Image Watermarking Model

The proposed code-based multiple watermarking model achieves efficient embedding and extraction performance while maintaining robustness and imperceptibility. Compared to the repetitive watermarking scheme, the proposed code-based approach exhibits comparable, and in several cases lower, embedding times. This improvement is mainly attributed to the finer partitioning strategy.

More specifically, in the repetitive watermarking scheme, a 4K image is partitioned





Figure 6.4: Experimental data-set for the evaluation of the repetitive SiP model.

into larger *cell* structures of approximate size  $167 \times 167$ , resulting in an embedding time of approximately 84 seconds due to the increased computational complexity of the performed FFT operations. In contrast, the proposed code-based model initially partitions the image into 36 *authenticity-blocks* and subsequently each *authenticity-block* is further divided through a  $9 \times 9$  *grid* into smaller *cells* of size approximately  $40 \times 40$ . As a result, the FFT computations are performed over significantly smaller regions, reducing the computational complexity of the frequency-domain processing and consequently decreasing the overall embedding time.

Therefore, the proposed partitioning strategy not only preserves the robustness and extraction capabilities of the watermarking process, but also improves the computational efficiency of the embedding procedure, making the proposed code-based multiple watermarking model more suitable for practical and large-resolution forensic imaging scenarios.

### ROMFAIA Hybrid Watermarking Model

Table 6.4 presents the computational cost results of our proposed scheme. Specifically, for each resolution group, the average execution times are reported for the

Table 6.4: Average execution times (in seconds) of the ROMFAIA watermarking model.

| Resolution Group | $T^{Em} * A$ | $T^{Em} * I$ | $T^{Em}$ | $T^{Ex} * A$ | $T^{Ex} * I$ | $T^{Ex}$ |
|------------------|--------------|--------------|----------|--------------|--------------|----------|
| 4K               | 33.95        | 5.96         | 39.91    | 33.47        | 1.85         | 35.32    |
| 2K               | 10.50        | 1.39         | 11.89    | 9.73         | 0.32         | 10.05    |
| 1080p            | 3.68         | 1.68         | 5.36     | 3.05         | 0.89         | 3.94     |
| 1024p            | 3.30         | 1.54         | 4.84     | 2.69         | 0.92         | 3.61     |
| 720p             | 2.00         | 0.80         | 2.80     | 1.72         | 0.46         | 2.18     |
| Randomized       | 3.27         | 0.57         | 3.84     | 2.67         | 0.23         | 2.90     |
| <b>Average</b>   | 9.45         | 1.99         | 11.44    | 8.89         | 0.78         | 9.67     |

embedding of authenticity and integrity information ( $T_A^{Em}$  and  $T_I^{Em}$ ), as well as for the corresponding extraction procedures ( $T_A^{Ex}$  and  $T_I^{Ex}$ ). Additionally, the total execution times for embedding and extraction ( $T^{Em}$  and  $T^{Ex}$ ) are provided. Finally, the last row of the table reports the overall average execution time across all resolution groups.

The results indicate that the proposed technique achieves efficient computational performance across all resolution groups. Notably, even for high-resolution images such as 4K, the total execution time for both embedding and extraction remains below 40 seconds, demonstrating the practical applicability of the method in real-world scenarios. Moreover, the execution times exhibit a clear correlation with image resolution. Higher-resolution images result in increased computational cost, while lower-resolution images (e.g., 720p) achieve significantly faster processing times. This behavior is expected, as the number of processed pixels directly affects the complexity of both embedding and extraction procedures. In addition, it is observed that the embedding of authenticity information ( $T_A^{Em}$ ) constitutes the dominant portion of the total embedding time compared to integrity embedding ( $T_I^{Em}$ ), due to the computational overhead introduced by the frequency-domain processing. In contrast, the integrity-related operations remain relatively lightweight, as they mainly rely on hash computations and LSB modifications. A similar trend is observed during the extraction phase, where authenticity extraction ( $T_A^{Ex}$ ) requires more computational effort

compared to integrity verification ( $T_I^{Ex}$ ). Despite this, the overall extraction time remains consistently lower than or comparable to the embedding time, indicating an efficient recovery procedure. Finally, the *Randomized* group presents intermediate execution times, falling between those of the *720p* and *1080p* groups. This is consistent with its composition, as it includes images with varying dimensions within this range, thus leading to proportionally moderate computational requirements.

Overall, the results confirm that the proposed scheme scales predictably with image resolution while maintaining a favorable balance between performance and computational efficiency.

### **ProtAIN Dual NFT watermarking Model**

The computational performance of the proposed ProtAIN model was evaluated by measuring the execution time of both its security-preservation and deployment procedures. More specifically, the average execution time required for the sequential embedding of the authenticity and integrity related information was estimated at 209 seconds, while the corresponding extraction and verification procedures required 139 seconds on average. The generation of the backup NFT image through the Arnold transformation required an average of 0.78 seconds. Furthermore, the deployment of each smart contract required approximately 0.014 seconds, whereas the complete deployment of the backup NFT, including the upload of the transformed image to IPFS and the registration of its corresponding smart contract on the blockchain, required approximately 5 seconds. Finally, the deployment of the original NFT, consisting of the upload of the protected image to IPFS and the deployment of its corresponding smart contract on the blockchain prior to its release on the marketplace, required approximately 7 seconds on average.

### **Cross-block Permutation-based Image Integrity-Preservation Model**

The run-time complexity of the proposed cross-block watermarking model is significantly lower compared to the previous authenticity preservation models operating in the frequency domain. More specifically, the computation of the cryptographic hash values for the corresponding *integrity-blocks* is a linear-time procedure depending on the total number of *integrity-blocks* processed during the embedding and verification stages. Furthermore, the generation of the permutation utilized for establishing the inter-*integrity-block* verification dependencies between the red and green channels

Table 6.5: Imperceptibility measurements of the proposed repetitive SiP watermarking model.

| Image      | Size               | PSNR  | SSIM |
|------------|--------------------|-------|------|
| people     | $4096 \times 4096$ | 33.35 | 0.84 |
| horizon    | $4096 \times 4096$ | 39.28 | 0.97 |
| lighthouse | $8192 \times 8192$ | 34.49 | 0.87 |
| elephant   | $8192 \times 8192$ | 38.42 | 0.96 |

is performed only once prior to the embedding procedure and therefore its computational overhead is not considered part of the effective embedding complexity. The corresponding LSB embedding process constitutes a constant-time operation for each *integrity-block*, since a fixed number of 256 bits is embedded into every verifier *integrity-block* according to the established permutation mapping.

Overall, both the embedding and verification procedures exhibit very low execution times, typically remaining below approximately 1 second for the evaluated forensic images. Consequently, the proposed model is highly suitable for scalable and near real-time integrity-preservation environments, where efficient tamper detection and localization must be achieved without introducing significant computational overhead.

## 6.5.2 Images: Quality and Imperceptibility

### Repetitive SiP Watermarking Model

Table 6.5 presents the PSNR and SSIM measurements obtained for the proposed repetitive SiP watermarking model. The experimental results demonstrate that the proposed methodology preserves high perceptual quality and visual fidelity while embedding the watermark information into high-resolution images.

More specifically, the obtained PSNR values range from 33.35 dB to 39.28 dB, indicating that the distortions introduced by the embedding procedure remain visually imperceptible. Similarly, the SSIM values range from 0.84 to 0.97, demonstrating a high degree of structural similarity between the original and the corresponding watermarked images.

The results further indicate that the proposed model maintains stable imperceptibility characteristics across different image contents and resolutions. Even for ultra-

Table 6.6: Imperceptibility evaluation of the image code-based multiple watermarking model.

| Image            | PSNR (dB)    | SSIM         |
|------------------|--------------|--------------|
| Forensic Image 1 | 41.05        | 0.95         |
| Forensic Image 2 | 41.08        | 0.95         |
| Forensic Image 3 | 41.08        | 0.95         |
| Forensic Image 4 | 41.17        | 0.95         |
| Forensic Image 5 | 41.08        | 0.96         |
| <b>Average</b>   | <b>41.09</b> | <b>0.952</b> |

high-resolution images of dimensions up to  $8192 \times 8192$ , the watermarking process introduces only limited visual distortion while preserving the structural properties of the host image. Overall, the obtained measurements confirm the effectiveness of the proposed repetitive SiP watermarking model in achieving a favorable balance between watermark embedding and perceptual quality preservation.

#### Code-based Multiple SiP Image Watermarking Model

Table 6.6 presents the achieved PSNR and SSIM values after the embedding of the code-sequence into the forensic images. As can be observed from the experimental results, the proposed embedding process introduces only minimal visual distortion despite embedding more SiP sequences, preserving the perceptual quality and structural characteristics of the original forensic images. More specifically, in all examined cases the achieved PSNR values remain above 41 dB, indicating a high level of imperceptibility of the embedded information, while the obtained SSIM values are consistently close to 1, demonstrating a very high structural similarity between the original and the watermarked forensic images. Overall, the proposed approach achieves an average PSNR value of 41.09 dB and an average SSIM value of 95.2%, highlighting the fidelity and transparency of the embedding scheme while maintaining the visual integrity of the forensic content.

Table 6.7: Average performance results per resolution group, reporting PSNR (dB), SSIM, Extraction-Rate  $R_e$  (%), BER, and tamper detection outcomes, computed over all five images per resolution group for the ROMFAIA model.

| Resolution Group | PSNR   | SSIM   | $R_e$  | BER    | Tampered |
|------------------|--------|--------|--------|--------|----------|
| 4K               | 53.012 | 0.9956 | 100.00 | 0.0000 | $\times$ |
| 2K               | 37.055 | 0.9826 | 100.00 | 0.0000 | $\times$ |
| 1080p            | 46.614 | 0.9897 | 100.00 | 0.0000 | $\times$ |
| 1024p            | 33.095 | 0.9194 | 100.00 | 0.0000 | $\times$ |
| 720p             | 34.509 | 0.9600 | 100.00 | 0.0000 | $\times$ |
| Randomized       | 40.511 | 0.9771 | 100.00 | 0.0000 | $\times$ |
| <b>Average</b>   | 40.808 | 0.9707 | 100.00 | 0.0000 | –        |

### ROMFAIA Hybrid Watermarking Model

The results presented in Table 6.7 demonstrate the effectiveness of the proposed watermarking system in preserving the visual quality of medical images across different resolution groups. In particular, the PSNR values remain overall high, with an average of 40.808 dB, indicating that the embedding of authenticity and integrity information introduces negligible perceptual distortion, although variations are observed across resolutions. Similarly, the SSIM values are consistently close to 1, with a general average of 0.9707, confirming that the structural characteristics of the medical images are well preserved.

A closer examination of the results shows that higher-resolution images yield the highest visual quality. Specifically, the 4K resolution attains a PSNR of 53.012 dB and an SSIM of 0.9956, while the 1080p resolution achieves 46.614 dB and 0.9897, respectively. In contrast, mid- and lower-resolution images exhibit reduced values. The 2K resolution records 37.055 dB and 0.9826, while the 1024p case drops to 33.095 dB and 0.9194. Similarly, the 720p resolution achieves 34.509 dB and 0.9600. The randomized case yields intermediate performance, with a PSNR of 40.511 dB and an SSIM of 0.9771. This behavior can be attributed to the relatively stronger impact of embedding modifications on smaller image regions, where each modification affects a larger proportion of the local structure.

Furthermore, the Extraction-Rate  $R_e$  achieves a perfect score of 100% across all

Table 6.8: Imperceptibility evaluation of the image cross-block based on permutation watermarking model.

| Image            | PSNR (dB)    | SSIM        |
|------------------|--------------|-------------|
| Forensic Image 1 | 55.91        | 0.99        |
| Forensic Image 2 | 55.91        | 0.99        |
| Forensic Image 3 | 55.89        | 0.99        |
| Forensic Image 4 | 55.91        | 0.99        |
| Forensic Image 5 | 55.93        | 0.99        |
| Forensic Image 6 | 55.89        | 0.99        |
| <b>Average</b>   | <b>55.90</b> | <b>0.99</b> |

cases, demonstrating the reliability of the proposed method in accurately retrieving the embedded authenticity information under non-attacked conditions. In addition, the BER remains equal to 0 for all cases, indicating error-free extraction of the embedded data. No false tamper detections are observed, confirming that the integrity mechanism does not introduce false positives in clean images.

Overall, these findings confirm that the proposed watermarking system preserves high visual fidelity while ensuring accurate and distortion-free embedding of authenticity and integrity information under non-attacked conditions.

### ProtAIN Dual NFT watermarking Model

Since the proposed ProtAIN framework directly extends the ROMFAIA model without modifying its authenticity and integrity embedding procedures, its perceptual quality remains consistent with that of the underlying framework. Accordingly, the results presented in Table 6.8 are also representative of the protected NFT images generated by the proposed methodology. More specifically, the sequential embedding of the authenticity and integrity information yields an average PSNR of 40.81 dB and an average SSIM of 0.971, while the embedded authenticity information is recovered with an Extraction Rate of 100% under no-attack conditions. These results indicate that the integration of the NFT deployment mechanism does not introduce additional visual distortion beyond the underlying watermarking framework, thereby preserving the visual quality of the protected NFT images.

### Cross-block Permutation-based Image Integrity-Preservation Model

Table 6.8 presents the achieved PSNR and SSIM measurements for the evaluation of the imperceptibility and fidelity of the forensic images after the deployment of the proposed information hiding procedure. As observed from the obtained results, the PSNR values remain above 50 dB, while the corresponding SSIM measurements remain very close to 1.0, demonstrating that the proposed model preserves both the perceptual quality and the structural similarity of the protected forensic images. The obtained measurements indicate that the modifications introduced by the embedding procedure remain visually imperceptible and do not significantly affect the visual content of the protected images. These results are particularly significant considering that the payload introduced for storing the integrity information is relatively high. More specifically, the proposed model utilizes 1024 *integrity-blocks*, where each *integrity-block* stores a 256-bit cryptographic hash value, resulting in an effective embedding payload of:

$$1024 \times 256 = 262,144 \text{ bits}$$

Despite the high embedding capacity required for the integrity-preservation procedure, the obtained PSNR and SSIM measurements remain at very high levels. This behavior is mainly attributed to the deployment of LSB-based embedding, where only the least significant bits of the pixel values are modified during the information hiding procedure. Consequently, the introduced modifications remain visually imperceptible and do not significantly affect either the perceptual quality or the structural similarity of the protected forensic images.

### 6.5.3 Images: Authenticity Verification

#### Repetitive SiP Watermarking Model

As we can observe from the results depicted in Table 6.9, regarding the case of *Vertical-Side* crop, where the image is cropped vertically from one side, all images regardless of the cropping percentage achieve a successful extraction of the embedded watermark in both the technique proposed in [1] and the repetitive watermarking technique proposed in this work. Similarly, for the case of *Horizontal-Side* crop, where the image is cropped horizontally from one side, both techniques achieve successful watermark extraction for cropping percentages of 25% and 50%, while for the case of



Table 6.9: Extraction results of the repetitive SiP model in comparison with the basic SiP approach [1] under different cropping attacks.

| Image      | Crop (%) | Vertical-Side |                | Horizontal-Side |                | Vertical-Intermediate |                | Horizontal-Intermediate |                |
|------------|----------|---------------|----------------|-----------------|----------------|-----------------------|----------------|-------------------------|----------------|
|            |          | Basic SiP [1] | Repetitive SiP | Basic SiP [1]   | Repetitive SiP | Basic SiP [1]         | Repetitive SiP | Basic SiP [1]           | Repetitive SiP |
| people     | 25%      | ✓             | ✓              | ✓               | ✓              | ✗                     | ✓              | ✓                       | ✓              |
| horizon    | 25%      | ✓             | ✓              | ✓               | ✓              | ✗                     | ✓              | ✓                       | ✓              |
| lighthouse | 25%      | ✓             | ✓              | ✓               | ✓              | ✗                     | ✓              | ✓                       | ✓              |
| elephant   | 25%      | ✓             | ✓              | ✓               | ✓              | ✗                     | ✓              | ✓                       | ✓              |
| people     | 50%      | ✓             | ✓              | ✓               | ✓              | ✗                     | ✓              | ✓                       | ✓              |
| horizon    | 50%      | ✓             | ✓              | ✓               | ✓              | ✗                     | ✓              | ✓                       | ✓              |
| lighthouse | 50%      | ✓             | ✓              | ✓               | ✓              | ✗                     | ✓              | ✓                       | ✓              |
| elephant   | 50%      | ✓             | ✓              | ✓               | ✓              | ✗                     | ✓              | ✓                       | ✓              |
| people     | 75%      | ✓             | ✓              | ✗               | ✓              | ✓                     | ✓              | ✓                       | ✓              |
| horizon    | 75%      | ✓             | ✓              | ✗               | ✓              | ✗                     | ✓              | ✓                       | ✓              |
| lighthouse | 75%      | ✓             | ✓              | ✗               | ✓              | ✓                     | ✓              | ✓                       | ✓              |
| elephant   | 75%      | ✓             | ✓              | ✗               | ✓              | ✓                     | ✓              | ✓                       | ✓              |

75% cropping the technique proposed in [1] fails to extract the embedded watermark successfully.

Moreover, considering the cases of *Vertical-Intermediate* and *Horizontal-Intermediate* crop, the results presented in Table 6.9 demonstrate that for the *Vertical-Intermediate* crop, where the image is cropped vertically in an intermediate region, the technique proposed in [1] fails to extract the embedded watermark in most cases, while the repetitive watermarking technique proposed in this work achieves successful watermark extraction in all cases, either by directly extracting the complete watermark or by reconstructing it from the extracted watermark fragments. On the other hand, for the case of *Horizontal-Intermediate* crop, where the image is cropped horizontally in an intermediate region, both the technique proposed in [1] and the repetitive watermarking technique proposed in this work achieve successful watermark extraction in all examined cases.

At this point, it should be noted that the effectiveness of the repetitive watermarking technique is demonstrated by the fact that successful watermark extraction is achieved in all cropping scenarios regardless of the cropping percentage, including the extreme case where only 25% of the original image remains available to contain the hidden information.

On the other hand, considering the compression attacks, we distinguish three different compression quality factors applied to the watermarked image, namely 90%, 80%, and 70%. The corresponding extraction results, together with the exhibited *PSNR*

Table 6.10: Extraction results of the repetitive model in comparison with the basic SiP approach [1] under JPEG compression attacks with different quality factors.

| Image      | Quality = 90% |      |      |                |      |      | Quality = 80% |       |      |                |       |      | Quality = 70% |      |      |                |      |      |
|------------|---------------|------|------|----------------|------|------|---------------|-------|------|----------------|-------|------|---------------|------|------|----------------|------|------|
|            | Basic SiP [1] |      |      | Repetitive SiP |      |      | Basic SiP [1] |       |      | Repetitive SiP |       |      | Basic SiP [1] |      |      | Repetitive SiP |      |      |
|            | Ext.          | PSNR | SSIM | Ext.           | PSNR | SSIM | Ext.          | PSNR  | SSIM | Ext.           | PSNR  | SSIM | Ext.          | PSNR | SSIM | Ext.           | PSNR | SSIM |
| people     | ✓             | 33.7 | 0.87 | ✓              | 33.2 | 0.84 | ✓             | 33.4  | 0.86 | ✓              | 33.01 | 0.82 | ✓             | 33.2 | 0.85 | ✓              | 32.8 | 0.82 |
| horizon    | ✓             | 40.4 | 0.96 | ✓              | 39.0 | 0.96 | ✓             | 40.02 | 0.96 | ✓              | 38.7  | 0.96 | ✓             | 39.5 | 0.96 | ✓              | 38.3 | 0.95 |
| lighthouse | ✓             | 34.5 | 0.88 | ✓              | 34.4 | 0.86 | ✓             | 34.5  | 0.87 | ✓              | 34.3  | 0.86 | ✗             | 34.4 | 0.87 | ✓              | 34.3 | 0.86 |
| elephant   | ✓             | 39.2 | 0.96 | ✓              | 38.3 | 0.95 | ✗             | 39.1  | 0.96 | ✓              | 38.2  | 0.95 | ✗             | 39.1 | 0.96 | ✓              | 37.9 | 0.95 |

and *SSIM* values, are presented in Table 6.10. In all examined cases, we evaluate the successful extraction of the embedded watermark through both the initial watermarking technique proposed in [1] and the repetitive watermarking scheme proposed in this work.

As we can observe from the experimental results presented in Table 6.10, the proposed repetitive watermarking scheme preserves high imperceptibility and robustness against JPEG compression attacks, even in cases of stronger compression corresponding to lower image quality. More specifically, the achieved PSNR and SSIM values remain comparable to those obtained by the baseline technique proposed in [1], while the proposed repetitive watermarking scheme demonstrates improved robustness in terms of successful watermark extraction. In particular, for lower compression quality factors, i.e., 80% and 70%, the baseline technique fails to extract the embedded watermark successfully in several cases, whereas the repetitive watermarking scheme proposed in this work achieves successful watermark extraction in all examined cases.

Finally, Table 6.11 presents the percentages of successful watermark extraction cases after the application of the corresponding attacks. More specifically, the percentages depicted in Table 6.11 represent the number of cases where each watermarking technique achieved successful extraction of  $w$ , either by directly extracting the complete embedded watermark or by reconstructing it through the extracted watermark segments. Comparing the experimental results obtained through the evaluation of the proposed repetitive watermarking model for digital images against the baseline approach, it can be observed from Table 6.11 that the repetitive watermarking scheme proposed in this work achieves 100% successful watermark extraction in all examined attack scenarios. In contrast, the baseline technique proposed in [1] fails in several cases to reconstruct successfully the embedded watermark from the extracted

Table 6.11: Comparative results after crop and compression attacks.

|                | Vertical-Side | Horizontal-Side | Vertical-Intermediate | Horizontal-Intermediate | Comp. 90% | Comp. 80% | Comp. 70% |
|----------------|---------------|-----------------|-----------------------|-------------------------|-----------|-----------|-----------|
| Basic SiP [1]  | 100%          | 25%             | 100%                  | 66%                     | 100%      | 75%       | 50%       |
| Repetitive SiP | 100%          | 100%            | 100%                  | 100%                    | 100%      | 100%      | 100%      |

Table 6.12: Code extraction rate  $R_e$  (%) for different forensic images under various attacks for the evaluation of robustness of the code-based multiple watermarking model.

| Image            | S&P    | G_BL  | G_BL  | HEQ | CMP      | CMP      | CMP      |
|------------------|--------|-------|-------|-----|----------|----------|----------|
|                  | (0.05) | (3×3) | (5×5) | –   | (Q = 75) | (Q = 65) | (Q = 55) |
| Forensic Image 1 | 87.50  | 100   | 100   | 100 | 100      | 100      | 93.75    |
| Forensic Image 2 | 100.00 | 100   | 100   | 100 | 100      | 93.75    | 93.75    |
| Forensic Image 3 | 93.75  | 100   | 100   | 100 | 100      | 100      | 100      |
| Forensic Image 4 | 68.75  | 100   | 100   | 100 | 100      | 100      | 87.50    |
| Forensic Image 5 | 81.25  | 93.75 | 93.75 | 100 | 93.75    | 87.50    | 81.25    |
| Average          | 86.25  | 98.75 | 98.75 | 100 | 98.75    | 96.25    | 91.25    |

segments after the corresponding attack, highlighting the robustness improvement achieved through the proposed repetitive watermarking approach.

### Code-based Multiple SiP Image Watermarking Model

Table 6.12 presents the obtained code extraction rates  $R_e$  for different forensic images under various attack categories and attack factors after the application of the proposed watermarking technique. As can be observed from the experimental results presented in Table 6.12, the proposed approach demonstrates high robustness against a wide range of image processing attacks. In particular, for the *Salt and Pepper* attack, the code extraction rate  $R_e$  achieves an average value of 86.25%, with a minimum value of 68.75% and a maximum value of 100% across the forensic images of the employed data-set. Regarding the *Gaussian Blur* attack, for both the (3×3) and (5×5) kernels, the proposed technique achieves an average code extraction rate  $R_e$  of 98.75%, exhibiting values ranging from 93.75% up to 100%.

Furthermore, in the case of the *Histogram Equalization* attack, the proposed approach achieves a code extraction rate  $R_e$  equal to 100% for all examined forensic

images, demonstrating complete preservation of the embedded information after the corresponding attack. Finally, considering the *Compression* attack, the achieved extraction performance remains consistently high even for lower compression quality factors. More specifically, for compression quality factor equal to 75, the proposed technique achieves an average code extraction rate  $R_e$  of 98.75%, with values ranging between 93.75% and 100%. For compression quality factor equal to 65, the average extraction rate decreases slightly to 96.25%, exhibiting a minimum value of 87.5% and a maximum value of 100%. Similarly, for compression quality factor equal to 55, the achieved average extraction rate remains at adequately high levels with a mean value of 91.25%, ranging from 81.25% to 100%.

Overall, the obtained experimental results demonstrate that the proposed code-based multiple watermarking technique preserves high robustness and reliable extraction performance even under intensive image processing attacks and strong compression conditions.

### ROMFAIA Hybrid Watermarking Model

As shown in Table 6.13, the proposed method demonstrates consistent behavior under cropping operations across all resolution groups. The Extraction-Rate  $R_e$  decreases proportionally with the severity of the crop, with average values of approximately 75%, 50%, and 25% for 25%, 50%, and 75% cropping, respectively. Correspondingly, the BER increases progressively with attack severity, reaching average values of approximately 0.090, 0.177, and 0.258. This behavior is expected, as cropping removes portions of the image that contain embedded watermark information. Nevertheless, for moderate cropping levels, a substantial part of the embedded information remains recoverable across all resolution groups. Importantly, tampering is successfully detected in all cases, indicating that the integrity mechanism reliably identifies partial content removal regardless of cropping direction or severity.

Additionally, Table 6.14 presents the robustness results against common image processing operations. The proposed method is highly resilient to noise-based and sharpening operations, achieving near-perfect Extraction-Rate  $R_e$  values above 98% and negligible BER values. For example, Gaussian noise at all tested variance levels results in average  $R_e = 99.58\%$  and  $BER = 0.0003$ , while sharpening achieves  $R_e = 98.75\%$  and  $BER = 0.0009$ . Histogram equalization and gamma correction also preserve high extraction performance, with average  $R_e$  values of 96.67% and 85.63%,

Table 6.13: Average robustness performance of the ROMFAIA model under cropping attacks.

| Crop Type       | 25% Crop         |                  | 50% Crop         |                  | 75% Crop         |                  |
|-----------------|------------------|------------------|------------------|------------------|------------------|------------------|
|                 | $\overline{R_e}$ | $\overline{BER}$ | $\overline{R_e}$ | $\overline{BER}$ | $\overline{R_e}$ | $\overline{BER}$ |
| Top             | 75.00            | 0.0972           | 50.00            | 0.1771           | 25.00            | 0.2743           |
| Bottom          | 74.58            | 0.0802           | 49.79            | 0.1772           | 24.79            | 0.2582           |
| Left            | 74.58            | 0.0910           | 49.79            | 0.1873           | 24.79            | 0.2713           |
| Right           | 75.00            | 0.0833           | 50.00            | 0.1667           | 25.00            | 0.2639           |
| Both Horizontal | 75.00            | 0.0419           | 50.00            | 0.1771           | 25.00            | 0.2274           |
| Both Vertical   | 74.79            | 0.0529           | 50.00            | 0.1736           | 25.00            | 0.2290           |

Table 6.14: Average robustness performance of the proposed ROMFAIA model under filtering and intensity-transformation attacks.

| Attack Type                    | $\overline{R_e}$ (%) | $\overline{BER}$ |
|--------------------------------|----------------------|------------------|
| Gaussian Blur ( $3 \times 3$ ) | 71.83                | 0.0521           |
| Median Blur                    | 80.21                | 0.0260           |
| Mean Filter ( $3 \times 3$ )   | 70.42                | 0.0568           |
| Mean Filter ( $5 \times 5$ )   | 58.33                | 0.0767           |
| Sharpening Filter              | 98.75                | 0.0009           |
| Gaussian Noise (0.05)          | 99.58                | 0.0003           |
| Gaussian Noise (0.1)           | 99.58                | 0.0003           |
| Gaussian Noise (0.2)           | 99.58                | 0.0003           |
| Gamma Correction               | 85.63                | 0.0208           |
| Histogram Equalization         | 96.67                | 0.0033           |

respectively

In contrast, smoothing filters, particularly mean filtering with larger kernels, have

Table 6.15: Robustness evaluation of the proposed ROMFAIA model under JPEG compression attacks.

| Resolution Group | $Q = 90$ |        | $Q = 70$ |        | $Q = 50$ |        |
|------------------|----------|--------|----------|--------|----------|--------|
|                  | $R_e$    | BER    | $R_e$    | BER    | $R_e$    | BER    |
| $4K$             | 98.75    | 0.0014 | 85.00    | 0.0205 | 81.25    | 0.0305 |
| $2K$             | 98.75    | 0.0003 | 92.50    | 0.0083 | 63.75    | 0.0503 |
| $1080p$          | 96.25    | 0.0049 | 88.75    | 0.0309 | 71.25    | 0.0597 |
| $1024p$          | 100.00   | 0.0000 | 82.50    | 0.0288 | 46.25    | 0.0934 |
| $720p$           | 97.50    | 0.0021 | 83.75    | 0.0222 | 28.75    | 0.1310 |
| Randomized       | 96.25    | 0.0021 | 86.25    | 0.0198 | 60.00    | 0.0722 |
| <b>Average</b>   | 97.92    | 0.0018 | 86.46    | 0.0218 | 58.54    | 0.0532 |

a more significant impact. Mean filtering ( $3 \times 3$ ) reduces the average  $R_e$  to 70.42%, while mean filtering ( $5 \times 5$ ) further decreases Extraction-Rate  $R_e$  to 58.33%, with BER values of 0.0568 and 0.0767, respectively. This behavior is consistent with the embedding mechanism, as such filters attenuate high frequency components and reduce the contrast between the red and blue annuli used for watermark extraction. At the same time, tamper detection remains consistently successful across all cases, demonstrating the reliability of the integrity verification mechanism.

Continuing, the results in Table 6.15 indicate that the proposed watermarking scheme maintains strong robustness under compression, especially for high-quality factors. For  $Q = 90$ , the  $R_e$  reaches 97.92%, with a very low BER of 0.0018. When the compression quality decreases to  $Q = 70$ , the average  $R_e$  value remains relatively high at 86.46%, with BER = 0.0218. As compression becomes more aggressive, such as  $Q = 50$ , a noticeable degradation is observed, with the average  $R_e$  metric decreasing to 58.54% and BER increasing to 0.0532, particularly for lower-resolution images such as  $720p$ . This degradation is attributed to quantization effects that suppress high-frequency components where the watermark information is embedded. However, even under stronger compression, the watermark remains partially recoverable, and tampering is correctly detected in all cases. These findings highlight the

Table 6.16: Average robustness performance of the proposed ROMFAIA model under geometric attacks.

| Attack Type            | $R_e$ (%) | BER    |
|------------------------|-----------|--------|
| Scaling 110%           | 95.42     | 0.0050 |
| Scaling 90%            | 89.58     | 0.0121 |
| Translation (+5, +5)   | 91.46     | 0.0082 |
| Translation (+10, +10) | 67.92     | 0.0270 |

robustness of the method against lossy compression, which is common in practical medical imaging workflows.

Finally, Table 6.16 summarizes the performance under geometric transformations. The method demonstrates high robustness to scaling operations, with average  $R_e$  values of 95.42% and 89.58% for scaling factors of 110% and 90%, respectively. Small translations, such as (+5, +5), are also well tolerated, achieving an average  $R_e$  value of 91.46% and BER of 0.0082. However, larger translations, such as (+10, +10), introduce more significant degradation, reducing the average  $R_e$  value to 67.92% and increasing the BER to 0.0270, while still remaining within acceptable performance levels. This effect is especially visible in the 720p resolution group, where translation (+10, +10) reduces  $R_e$  to 22.50%. This sensitivity arises from spatial misalignment of the *grid*, which affects the correct identification of marked *cells* during extraction. Additionally, the results reveal a clear dependency on image resolution, with higher-resolution images exhibiting more stable robustness, while lower-resolution images are more sensitive to spatial distortions. Despite these challenges, tampering is successfully detected in all cases.

Overall, the experimental results demonstrate that the proposed watermarking system achieves a strong balance between robustness and sensitivity. The authenticity watermark remains resilient under several common image processing operations, including noise, sharpening, histogram equalization, moderate compression, and scaling. In contrast, stronger cropping, smoothing, and larger translations reduce extraction performance due to information loss, frequency attenuation, or spatial misalignment, while our scheme still maintains satisfactory robustness under these more challenging conditions. At the same time, the fragile integrity mechanism consistently

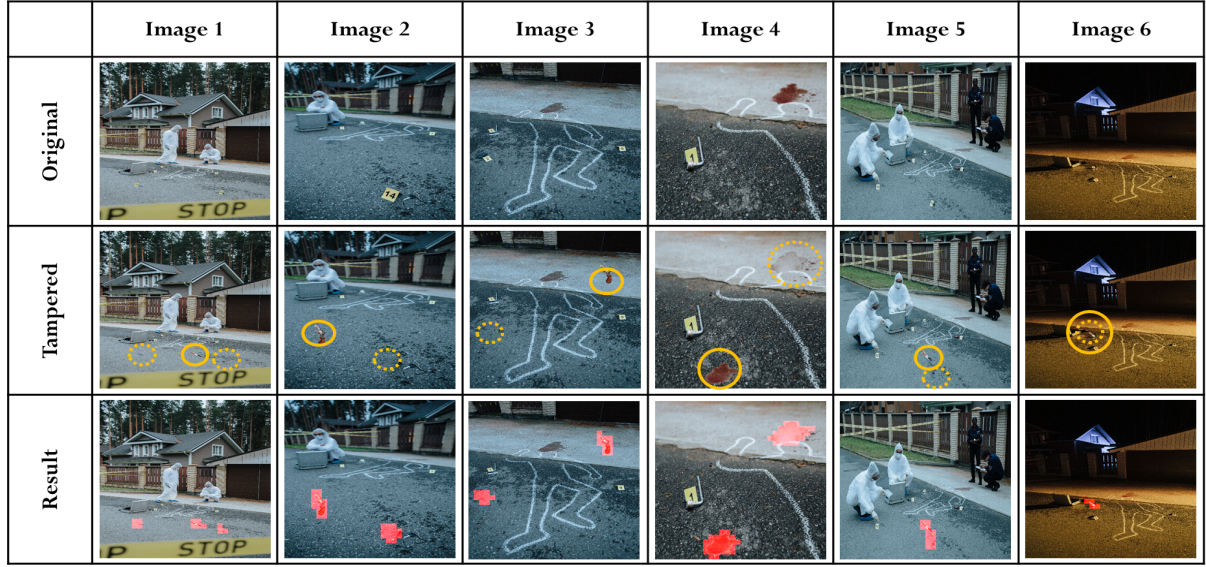


Figure 6.5: Tamper detection and localization visualization results after evaluating our model for the preservation of integrity in forensic images .

detects tampering across all evaluated attacks, confirming the complementary role of the robust authenticity and fragile integrity components in the proposed ROMFAIA system.

### ProtAIN Dual NFT Watermarking Model

The authenticity-preservation capabilities of the proposed *ProtAIN* framework are directly inherited from the underlying *ROMFAIA* model, as the frequency-domain watermark embedding and extraction procedures remain unchanged. Consequently, the robustness of the embedded watermark against common signal-processing and geometric attacks is identical to that demonstrated by the underlying framework. More specifically, the proposed methodology preserves high Extraction Rates under JPEG compression, filtering, geometric transformations, and cropping attacks, enabling reliable authenticity verification even after the protected NFT image has undergone common processing operations. Therefore, the integration of blockchain and IPFS mechanisms extends the deployment and protection of the digital asset without affecting the robustness characteristics of the embedded authenticity watermark.



### 6.5.4 Images: Integrity Attestation

#### Cross-block Permutation-based Image Integrity-Preservation Model

Figure 6.5 presents the corresponding tamper detection and localization results for the 6 forensic images utilized in the experimental evaluation. More specifically, the first row illustrates the watermarked forensic images after the deployment of the proposed integrity-preservation model. The second row presents the tampered forensic images. In these examples, the manipulated regions are highlighted using yellow circles, where continuous circles denote insertion attacks and dotted circles denote removal attacks. Finally, the third row presents the output of the proposed tamper detection and localization procedure. As observed from the results, the tampered regions are successfully localized through the annotation of the affected *integrity-blocks* using semi-transparent red overlays. More specifically, the marked *integrity-blocks* correspond to regions whose recomputed red-channel hash values fail to satisfy the established inter-*integrity-block* verification relations.

The experimental results demonstrate that the proposed model achieves approximately 100% Accuracy, Precision, Recall, and F1-score during the tamper detection and localization procedure, while simultaneously exhibiting nearly 0% false-positive detections. The low number of false-positive detections is mainly attributed to the deployment of the proposed  $\Lambda$ -relations between the corresponding *integrity-blocks*, which establish interconnected verification dependencies among the embedded integrity relations. More specifically, the  $\Lambda$ -relation mechanism prevents isolated verification mismatches from incorrectly identifying untampered *integrity-blocks* as manipulated, since a candidate *integrity-block* must simultaneously fail both the forward and backward integrity-verification relations in order to be characterized as tampered. Consequently, the proposed model enables accurate and reliable tamper detection and localization for forensic digital images while preserving high perceptual quality.

#### ROMFAIA Hybrid Watermarking Model

To evaluate the integrity preservation capability of the proposed scheme, we apply a set of insertion, removal, and combined attacks on the medical images of our dataset. The evaluation focuses on two key aspects: (i) the ability of the system to detect whether a medical image  $I_{rf}$  has been tampered with, and (ii) its ability to accurately

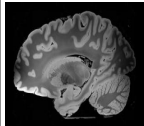
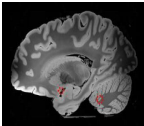
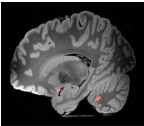
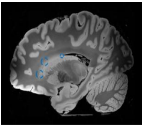
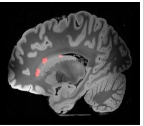
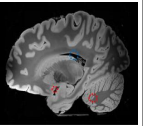
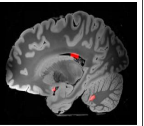
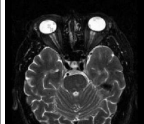
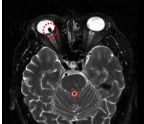
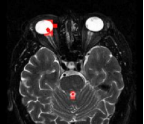
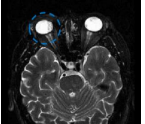
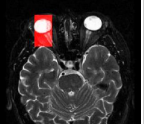
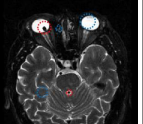
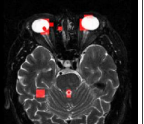
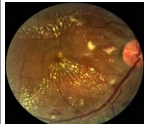
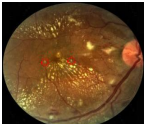
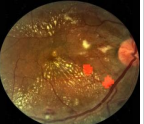
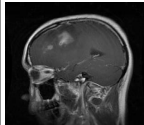
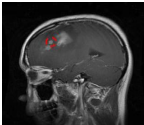
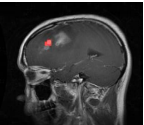
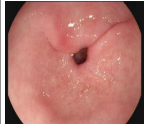
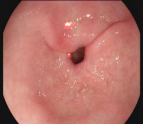
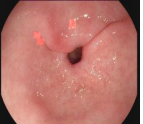
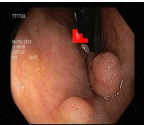
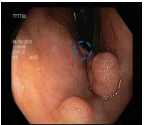
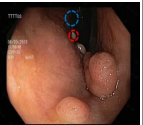
| Resolution Group | Original Image                                                                     | Insertion Attack                                                                   | Detection & Localization                                                           | Removal Attack                                                                     | Detection & Localization                                                            | Combined Attack                                                                      | Detection & Localization                                                             |
|------------------|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 4K               |   |   |   |   |   |   |   |
| 2K               |   |   |   |   |   |   |   |
| 1080p            |   |   |   |   |   |   |   |
| 1024p            |   |   |   |   |   |   |   |
| 720p             |   |   |   |   |   |   |   |
| Randomized       |  |  |  |  |  |  |  |

Figure 6.6: Tamper Detection and Localization under Different Attack Scenarios. Representative results across multiple resolution groups, illustrating insertion, removal, and combined attacks along with the corresponding detection and localization outputs. The highlighted regions indicate successfully identified tampered blocks, demonstrating the effectiveness of the proposed ROMFAIA system in accurately detecting and localizing modifications.

localize the affected regions (at block level).

Figure 6.6 presents representative tamper detection and localization results across different resolution groups. The figure includes the original watermarked images, the tampered images under different attack types, and the corresponding detection and localization outputs. The proposed method marks as tampered the *integrity-blocks* that violate the verification rule, highlighting them in red. For visualization purposes, in insertion and removal attacks, the tampered regions are additionally indicated using red (insertion) and blue (removal) circles, allowing a direct comparison between actual and detected tampered areas. As observed, in all insertion attack cases, the system

successfully detects the presence of tampering and accurately localizes the inserted regions, marking the affected *integrity-blocks*. Similarly, for removal attacks, all cases are correctly identified as tampered, and the missing or altered regions are precisely localized. For combined attacks, which include both insertion and removal performed within the same image, the proposed method consistently detects tampering across all cases. The results demonstrate the ability of the technique to accurately localize multiple modifications simultaneously, even when they differ in size, occur in distant and non-overlapping regions, or involve both small-scale and large-scale alterations. This includes cases of subtle modifications, extensive tampered areas, and multiple manipulations distributed across different parts of the medical image.

Overall, the results demonstrate that our proposed system achieves reliable tamper detection and precise localization, even under complex and diverse tampering scenarios.

### **ProtAIN Dual NFT Watermarking Model**

The integrity-preservation capabilities of the ProtAIN model are directly inherited from ROMFAIA model, as the integrity embedding and verification procedures remain unchanged. The embedded integrity information enables reliable identification and localization of image modifications under insertion, removal and combined attacks, through the detection of inconsistencies in the established inter-block verification relations. Consequently, the NFT deployment mechanism extends the underlying watermarking framework with blockchain-based asset protection while preserving its integrity verification and tamper-localization capabilities.

### **6.5.5 Images: Observations**

Overall, the experimental evaluation demonstrates that the proposed image-based watermarking models achieve an effective balance between robustness, imperceptibility, computational efficiency, and integrity preservation. More specifically, the frequency-domain watermarking models exhibited high robustness against geometric transformations, compression, filtering, and cropping attacks while simultaneously preserving high perceptual quality, as reflected by the consistently high PSNR and SSIM values obtained throughout the experiments.

The repetitive SiP watermarking model demonstrated significantly improved ro-

bustness against cropping attacks compared to the baseline SiP approach [1], achieving successful watermark extraction even when only a limited portion of the original image remained available. Furthermore, the repetitive embedding strategy substantially reduced the computational complexity of the embedding procedure through the deployment of smaller transform regions during the FFT computations.

Similarly, the proposed code-based multiple watermarking model achieved high extraction rates under a wide range of image-processing attacks, including filtering, histogram equalization, and JPEG compression. The obtained results indicate that the deployment of multiple embedded code sequences improves the robustness and reliability of the extraction procedure while preserving high imperceptibility characteristics.

The proposed cross-block integrity-preservation model demonstrated highly accurate tamper detection and localization capabilities for forensic digital images. More specifically, the deployment of cryptographic hash values, inter-*integrity-block* permutation mappings, and  $\Lambda$ -relation verification dependencies enabled accurate localization of manipulated regions while minimizing false-positive detections. Despite the relatively high embedding payload required for integrity preservation, the proposed LSB-based embedding strategy maintained high PSNR and SSIM values, preserving the perceptual quality of the protected images.

Building upon these contributions, the proposed ROMFAIA framework successfully integrated robust authenticity preservation and fragile integrity verification into a unified image-security architecture. The experimental evaluation demonstrated that the authenticity component remained highly robust under common image-processing attacks, while the integrity component accurately detected and localized manipulated image regions through cryptographic verification dependencies. Furthermore, the sequential deployment of the authenticity and integrity layers preserved high perceptual quality, demonstrating that the simultaneous protection of authenticity and integrity can be achieved without introducing noticeable visual degradation.

Finally, the proposed ProtAIN framework extended the ROMFAIA architecture to the NFT ecosystem by integrating blockchain and IPFS technologies with the underlying watermarking methodology. Since the authenticity and integrity embedding procedures remain unchanged, the robustness, imperceptibility, and tamper-detection capabilities of the underlying framework are fully preserved. Consequently, ProtAIN demonstrates that robust watermarking and cryptographic verification mechanisms

can be effectively combined with decentralized storage and blockchain infrastructures, providing an integrated solution for the protection, deployment, and verification of digital image assets in NFT environments.

In general, the obtained experimental results verify that the proposed image-based frameworks provide scalable and computationally efficient solutions for authenticity preservation, integrity attestation, tamper localization, and secure digital asset protection, demonstrating their suitability for both conventional multimedia applications and emerging blockchain-based environments.

## 6.6 Experimental Results on Audio

Following the presentation of experimental results on images, we proceed to showcase the experimental results regarding the proposed audio-based authenticity and integrity preservation models. Specifically, the evaluation focuses on the computational efficiency, auditory quality, authenticity verification capability, and tamper detection and localization performance of the proposed watermarking models under different attack categories and signal-processing conditions.

The following audio-based models are experimentally evaluated:

- **Audio Watermarking Model utilizing Dual-Layer Embedding**
- **Permutation-Based Cross-Audio Block Integrity-Preservation Model**

### **Audio Watermarking Model utilizing Dual-Layer Embedding - Methodology**

The experimental evaluation of the proposed multiple watermarking model for digital audio signals focuses on both the robustness and the imperceptibility of the proposed embedding methodology under various signal-processing attacks. More specifically, the evaluation investigates the capability of the proposed model to successfully extract the embedded watermarked image  $I_w$  and the watermark  $w$  after the deployment of attacks affecting the protected audio signal. In cases where the extracted watermark is partially damaged, the reconstruction capability provided through the SiP properties described in [1] is additionally evaluated.

For the experimental evaluation, the integer value 5 was utilized as watermark information, encoded through the SiP  $\{4, 6, 7, 1, 5, 2, 3\}$  and subsequently embedded into the *Lena* image of dimensions  $128 \times 128$ . The resulting watermarked image was

| Type                   | Crop Size | Original SiP    | Watermarked Image                                                                 | Watermarked Signal                                                                | Cropped Signal                                                                     | Extracted Image                                                                     | Extracted SiP   |
|------------------------|-----------|-----------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-----------------|
| No Crop                | 0 %       | [4,6,7,1,5,2,3] |  |  |  |  | [4,6,7,1,5,2,3] |
| Left-Side Crop         | 25 %      | [4,6,7,1,5,2,3] |  |  |  |  | FIXED           |
|                        | 50 %      | [4,6,7,1,5,2,3] |  |  |  |  | FIXED           |
|                        | 75 %      | [4,6,7,1,5,2,3] |  |  |  |  | CORRUPTED       |
|                        |           |                 |                                                                                   |                                                                                   |                                                                                    |                                                                                     |                 |
| Intermediate-Side Crop | 25 %      | [4,6,7,1,5,2,3] |  |  |  |  | FIXED           |
|                        | 50 %      | [4,6,7,1,5,2,3] |  |  |  |  | FIXED           |
|                        | 75 %      | [4,6,7,1,5,2,3] |  |  |  |  | CORRUPTED       |
| Right-Side Crop        | 25 %      | [4,6,7,1,5,2,3] |  |  |  |  | FIXED           |
|                        | 50 %      | [4,6,7,1,5,2,3] |  |  |  |  | FIXED           |
|                        | 75 %      | [4,6,7,1,5,2,3] |  |  |  |  | CORRUPTED       |

Figure 6.7: Crop attack demonstration for the evaluation of the dual-layer audio watermarking model.

then embedded into digital audio signals of duration ranging from approximately 30 seconds up to 1 minute. The employed audio data-set included three categories of audio signals, namely music-only tracks, mixed music-and-song tracks, and voice-only recordings.

The robustness evaluation considered three main attack categories, namely *Crop Attacks*, *Noise Attacks*, and *Down-sampling Attacks*. For the cropping attacks also depicted in Figure 6.7, the audio signal was cropped from the left side, the right side, and intermediate regions of the audio sequence. For the noise attacks, random noise with different deviation factors was inserted into the audio signal. Finally, for the down-sampling attacks, the sample rate of the audio signal was reduced to one-half, one-quarter, and one-eighth of its original value.

## Permutation-Based Cross-Audio Block Integrity-Preservation Model - Methodology

The experimental evaluation of the proposed audio integrity-preservation model fo-

cuses on the capability of the proposed model to accurately detect and localize malicious modifications affecting protected audio signals under different tampering scenarios. More specifically, the evaluation investigates the robustness of the proposed cross-audio block verification mechanism established through the permutation-based XOR integrity relations between the corresponding audio blocks. The employed experimental data-set consisted of digital audio signals including music-only tracks, mixed music-and-song recordings, and voice-only signals.

The experimental evaluation considered two main attack categories, namely *Crop Attacks* and *Noise Injection Attacks*. For the crop attacks, both large-scale and localized cropping operations were performed on the watermarked audio signals. More specifically, the attacks included trimming operations affecting several seconds of the audio track, as well as smaller cropping operations affecting specific audio samples and bit regions. These attacks were applied on different temporal regions of the audio signal in order to evaluate the capability of the proposed model to detect both extensive and localized tampering attempts. For the noise injection attacks, multiple signal-processing modifications were applied to the watermarked audio signals, including amplification operations, normalization filters, linear filtering operations, and fade-in/fade-out modifications.

### 6.6.1 Audio: Run-time Efficiency

#### Audio Watermarking Model utilizing Dual-Layer Embedding

Regarding the run-time efficiency of the proposed model, the overall embedding procedure can be separated into two distinct stages with different computational characteristics. The first stage corresponds to the watermarking of the image  $I$ , which constitutes the most computationally intensive part of the model, since it requires multiple FFT operations over the corresponding image *cells*. The computational complexity of this stage mainly depends on the number and dimensions of the deployed *cells*, as well as on the repeated frequency-domain transformations required during the embedding procedure. On the other hand, the second stage corresponds to the embedding of the watermarked image into the audio signal through LSB substitution. As discussed in the previous sections, the LSB embedding procedure can be considered a constant-time operation, since it only requires direct modification of the least significant bits of the corresponding *audio-samples* without additional transform-domain

Table 6.17: Experimental evaluation results of the proposed dual-layer audio watermarking model under crop, noise, and down-sampling attacks.

| Audio          | Attack Factor | Crop Attacks      |                   |                   | Noise Attacks |         | Down-sampling |         |
|----------------|---------------|-------------------|-------------------|-------------------|---------------|---------|---------------|---------|
|                |               | $I_w/w(\text{L})$ | $I_w/w(\text{I})$ | $I_w/w(\text{R})$ | PSNR          | $I_w/w$ | PSNR          | $I_w/w$ |
| music          | 25%           | ✓/✓               | ✓/✓               | ✓/✓               | 62.19         | ✓/✓     | 62.19         | ✓/✓     |
|                | 50%           | ✓/✓               | ✓/✓               | ✓/✓               | 62.19         | ✓/✓     | 62.19         | ✓/✓     |
|                | 75%           | ✗/✓               | ✗/✗               | ✗/✗               | 62.19         | ✓/✓     | 55.70         | ✓/✓     |
| piano          | 25%           | ✓/✓               | ✓/✓               | ✓/✓               | 66.54         | ✓/✓     | 66.54         | ✓/✓     |
|                | 50%           | ✓/✓               | ✓/✓               | ✓/✓               | 66.54         | ✓/✓     | 66.54         | ✓/✓     |
|                | 75%           | ✗/✓               | ✗/✗               | ✗/✗               | 66.54         | ✓/✓     | 60.08         | ✓/✓     |
| crowd          | 25%           | ✓/✓               | ✓/✓               | ✓/✓               | 66.54         | ✓/✓     | 66.54         | ✓/✓     |
|                | 50%           | ✓/✓               | ✓/✓               | ✓/✓               | 66.54         | ✓/✓     | 66.54         | ✓/✓     |
|                | 75%           | ✗/✓               | ✗/✗               | ✗/✗               | 66.54         | ✓/✓     | 60.08         | ✓/✓     |
| <b>Average</b> | –             | 66.6% / 77.7%     |                   |                   | 65.09         | 100%    | 63.09         | 100%    |

processing.

Overall, the complete embedding procedure requires approximately 1 minute for the evaluated combinations of audio signals and watermarked images. Consequently, despite the deployment of multiple watermarking layers, the proposed model preserves relatively low computational overhead, making it suitable for practical authenticity-preservation applications involving digital audio content.

### Permutation-Based Cross-Audio Block Integrity-Preservation Model

Regarding the computational efficiency of the proposed cross-audio block integrity-preservation model, its run-time characteristics follow similar trends to those observed for the image integrity-preservation model presented in Section 6.5.1. More specifically, the generation of the permutation utilized to establish the verification dependencies between audio blocks can be considered a pre-processing step and therefore does not significantly contribute to the effective embedding complexity.

Following the separation of the audio signal into the corresponding left- and



right-channel LSB sequences, the key-generation procedure is performed. This operation can be considered a linear-time process whose complexity depends on the total number of generated *integrity-blocks*. Similarly, the generation of the XOR-based integrity information for each left-channel *integrity-block* and its subsequent embedding into the corresponding right-channel *integrity-block*, according to the deployed permutation mapping, also constitutes a linear-time procedure with respect to the total number of processed *integrity-blocks*.

Consequently, the overall embedding and verification procedures exhibit linear complexity with respect to the number of audio blocks, making the proposed model suitable for large audio files and scalable integrity-preservation applications.

## 6.6.2 Audio: Quality and Imperceptibility

### Audio Watermarking Model utilizing Dual-Layer Embedding

As demonstrated by the experimental results presented in Table 6.17, the watermarked audio signals preserve high PSNR values even after the deployment of strong signal-processing attacks, indicating that the embedding procedure introduces only minimal perceptual distortion to the audio content. In attack-free scenarios, the achieved PSNR values are even higher, further confirming the imperceptibility of the proposed embedding strategy. The preservation of the auditory quality is mainly attributed to the deployment of the LSB-based embedding mechanism, since modifications applied to the least significant bits of audio samples produce very limited changes in the amplitude of the audio waveform and therefore remain imperceptible to the human auditory system. The total payload embedded into the audio signal is directly proportional to the size of the watermarked image  $I_w$ . Considering that the embedded image has dimensions  $128 \times 128$  pixels and each pixel is represented using 8 bits, the total embedding payload can be calculated as:

$$payload_s = 128 \times 128 \times 8$$

resulting in a total payload of 131,072 bits. Despite this relatively high embedding payload, the experimental results demonstrate that the proposed model preserves high perceptual quality while enabling reliable extraction of both the embedded watermarked image and the watermark information. Consequently, the proposed model successfully preserves the acoustic fidelity of the watermarked audio signals.

### Permutation-Based Cross-Audio Block Integrity-Preservation Model

The proposed permutation-based cross-audio block integrity-preservation model embeds verification information into the LSBs of the audio signal. Since only the least significant bit of selected audio samples is modified, the introduced alterations are expected to have a negligible impact on the perceptual quality of the protected audio signal, thereby preserving its acoustic fidelity.

The overall embedding payload depends on the number of generated *integrity-blocks* and the amount of verification information associated with each block. Nevertheless, LSB substitution is well suited for integrity-preservation applications, as it provides sufficient embedding capacity while maintaining high perceptual transparency. Consequently, the proposed framework achieves an effective balance between embedding capacity and imperceptibility without introducing noticeable audible distortions.

### 6.6.3 Audio: Authenticity Verification

#### Audio Watermarking Model utilizing Dual-Layer Embedding

As observed from the experimental results in Table 6.17, the proposed model achieves successful extraction of both the watermarked image  $I_w$  and the watermark  $w$  for all cropping attacks of magnitude 25% and 50%. In the more severe case of 75% cropping, where only 25% of the original signal remains available, successful extraction of the watermark  $w$  remains feasible for the left-side cropping case, while the extraction of both  $I_w$  and  $w$  fails in the intermediate-side and right-side cropping scenarios. Overall, the proposed model achieves approximately 66.6% successful extraction performance under all examined cropping conditions, while demonstrating particularly strong robustness against left-side cropping attacks.

For the case of noise attacks demonstrated in Table 6.17, the proposed framework achieves successful extraction of both the watermarked image  $I_w$  and the watermark  $w$  in all examined cases. More specifically, additive noise utilizing standard deviation factors  $\sigma^2 \in \{0.001, 0.01, 0.5\}$  was inserted into the audio signals, while the corresponding PSNR measurements achieved an average value of approximately 65.09 dB. Similarly, for the down-sampling attacks, the proposed model also achieves successful extraction of both the watermarked image  $I_w$  and the watermark  $w$  for all examined

Table 6.18: Tamper detection and localization results for the proposed permutation-based cross-audio block model under different attack categories.

| Attack Type                                                            | Tamper Detected | Tamper Localized |
|------------------------------------------------------------------------|-----------------|------------------|
| Large-Scale Audio Cropping                                             | ✓               | ✓                |
| Localized Audio Cropping (Samples and Bits)                            | ✓               | ✓                |
| Noise Injection Attack (Amplification)                                 | ✓               | ✓                |
| Noise Injection Attack (Multiple Signal-Processing Operations)         | ✓               | ✓                |
| Resampling Attack (Sample-Rate Modification and Non-Linear Resampling) | ✓               | ✗                |

sample-rate reduction factors achieving an average value of approximately 63.09 dB.

Overall, the obtained experimental results demonstrate that the proposed model preserves high robustness against common signal-processing attacks affecting digital audio signals while maintaining successful watermark extraction and reconstruction capabilities.

#### 6.6.4 Audio: Integrity Attestation

##### Permutation-Based Cross-Audio Block Integrity-Preservation Model

The tamper detection and localization results are presented in Table 6.18. As observed, the proposed model successfully detected and localized tampering under all examined cropping and noise injection attacks. More specifically, both large-scale and localized cropping attacks were accurately detected and localized, since the modifications affect only specific audio regions and consequently violate the corresponding XOR-based integrity relations. Similarly, all examined noise injection attacks, including amplification and combinations of signal-processing operations, were successfully detected and localized, as they modify the sample values without altering the underlying audio-block structure.

A different behavior was observed for the resampling attacks. Although the proposed model successfully detected the tempering, accurate localization was not possi-

ble. This is because resampling alters the number and positions of the audio samples, changing the boundaries and contents of the generated *integrity-blocks*. Consequently, while the established integrity relations are violated and tampering is detected, the modified block structure prevents the reliable identification of the exact tampered segment.

Overall, the experimental results demonstrate that the proposed model achieves reliable tamper detection under all examined attack categories and accurate localization whenever the original audio-block structure is preserved.

### 6.6.5 Audio: Observations

The experimental results reveal a clear trade-off between robustness and localization accuracy depending on the nature of the performed attack. For the dual-layer audio watermarking model, the deployment of the watermarked image as an intermediate carrier significantly improves robustness, since even when the extracted image is partially damaged, the embedded watermark can still be recovered or reconstructed through the SiP properties. This behavior is particularly evident under severe cropping attacks, where successful watermark extraction remains feasible despite substantial loss of audio content. Furthermore, both proposed models benefit from the deployment of LSB-based embedding, achieving high imperceptibility despite the relatively large embedding payloads. This confirms that modifications of the least significant bits introduce negligible perceptual distortion while still providing sufficient embedding capacity for authenticity and integrity information.

For the integrity-preservation model, the results demonstrate that attacks preserving the original audio-block structure, such as cropping and noise injection, can be accurately localized through the proposed permutation-based verification dependencies. In contrast, global transformations such as re-sampling modify the sample positions and consequently the generated *integrity-blocks*, preventing precise localization. Nevertheless, the framework remains capable of reliably detecting the integrity violation, indicating that the proposed verification mechanism is resilient even under severe structural modifications of the audio signal.

Overall, the obtained results suggest that the proposed models are particularly suitable for applications requiring high perceptual quality, reliable authenticity verification, and accurate detection of localized tampering attempts.

## 6.7 Experimental Results on Video

Next we present the experimental results regarding the proposed video authenticity and integrity preservation models. Specifically, the evaluation focuses on the computational efficiency, visual quality, authenticity verification, and tamper detection performance of the proposed watermarking models under different attack categories and signal-processing conditions.

The following video-based models are experimentally evaluated:

- **Code-based Hash Chain Video Watermarking Model**
- **Cross-Frame Permutation Based Video Watermarking Model**

### Code-based Hash Chain Video Watermarking Model - Methodology

To evaluate the proposed video model, we used VGG Sound dataset [71], that provides diverse audio–visual content and it fits very much to our audio–visual binding scheme. In Section 6.1 we provide an overview of the characteristics of the dataset we used in the experimental evaluation of our model. For the experimental evaluation, each video frame was partitioned using a  $4 \times 4$  *authenticity-plane*, resulting in 16 *authenticity-blocks* per frame. The watermark information consisted of a sequence of 16 integers in the range  $[8, 15]$ , corresponding to values whose binary representation requires 4 bits. Consequently, each *grid* was partitioned into  $9 \times 9$  *cells* to accommodate the embedding process. The selected configuration provides a balance between embedding capacity, perceptual quality, and robustness against attacks.

The robustness of the proposed model was evaluated against a diverse set of attacks commonly encountered during video acquisition, processing, compression, transmission, and redistribution. The selected attacks model both unintentional degradations, such as noise and filtering operations, and deliberate modifications that may affect watermark extraction and authenticity verification. More specifically, the evaluation includes filtering attacks (Average Blur, Gaussian Blur, and Median Blur), noise attacks (Gaussian Noise and Salt-and-Pepper Noise), compression attacks (JPEG and H.264 compression), geometric and temporal attacks (cropping, frame dropping, and frame swapping), as well as intensity-based transformations (Gamma Correction and Histogram Equalization). These attacks represent common video-processing operations encountered in practical multimedia environments.

Table 6.19: Evaluated attacks and corresponding parameters used for robustness assessment of the proposed code-based hash-chain video watermarking model.

| Attack Type            | Description                          | Parameters                                                         |
|------------------------|--------------------------------------|--------------------------------------------------------------------|
| Average Blur           | Spatial smoothing (mean filtering)   | Kernel = $3 \times 3$ , $5 \times 5$ , $7 \times 7$ , $9 \times 9$ |
| Gaussian Blur          | Low-pass filtering (Gaussian kernel) | Kernel = $9 \times 9$                                              |
| Median Blur            | Nonlinear filtering (impulse noise)  | Kernel = $3 \times 3$ , $9 \times 9$                               |
| Gaussian Noise         | Additive white Gaussian noise        | $\sigma = 0.04, 0.05, 0.4, 0.5$                                    |
| Salt-and-Pepper Noise  | Random pixel corruption              | Default impulse density                                            |
| JPEG Compression       | Lossy image compression              | Quality factor $Q = 50, 70$                                        |
| H.264 Compression      | Lossy video compression              | CRF = 22, 28                                                       |
| Cropping               | Frame-region removal                 | Top/Bottom, Left/Right, 25% – 50%                                  |
| Gamma Correction       | Nonlinear intensity transformation   | $\gamma = 1.25, 2.5$                                               |
| Histogram Equalization | Global contrast enhancement          | Standard histogram equalization                                    |

The evaluated attacks and their corresponding parameters are summarized in Table 6.19. Furthermore, the evaluation also considers combined attacks involving spatial scaling, frame-rate modification, and video re-encoding. Such compound transformations frequently occur when videos are uploaded to social media and video-sharing platforms, providing a realistic assessment of the robustness of the proposed watermarking model under practical operating conditions. All benchmark experiments were conducted on a Ryzen 7 5700X 8-core CPU with 32 GB RAM running Windows 10.

## Cross-Frame Permutation Based Video Watermarking Model - Methodology

To ensure the robustness and generalizability of the proposed model, we evaluate its performance across videos exhibiting diverse characteristics. For this purpose, we utilize the UCF Crime Dataset [72], which provides a collection of surveillance videos with different properties, including variations in scene complexity, motion dynamics, illumination, and recording conditions. From this data-set, we select a subset of 10 videos from two crime categories. For more details of the data-set characteristics see Table 6.2.

Each tested video frame was partitioned into a  $15 \times 20$  *grid*, resulting in *integrity-blocks* of size  $16 \times 16$  pixels. This configuration provides exactly 256 pixels per block, allowing direct embedding of a 256-bit hash without the need for spreading or step-size allocation. The permutation for each video is defined over  $F \cdot B$  elements, where  $F$  denotes the total number of frames and  $B$  the number of *integrity-blocks* per frame (in this case,  $B = 300$ ). Since the video data-set consists of videos with varying durations, a distinct permutation was generated for each video.

For the evaluation of the proposed model, several categories of content modification attacks were considered. Visual insertion and manipulation attacks were performed through the addition of external objects into selected video frames, while audio manipulation attacks were conducted by replacing portions of the original audio stream with audio segments originating from different videos. Furthermore, content-aware object removal attacks were generated using AI-based inpainting techniques, including LaMa-based pipelines, enabling realistic visual modifications with minimal perceptual artifacts. The evaluation also included combined attack scenarios involving simultaneous modifications of both the audio and visual modalities. In addition temporal manipulations were performed through localized and frame sequence alterations. Temporal manipulations included frame reordering operations, while preserving audio-video synchronization, in order to simulate realistic multimedia editing and content manipulation. Collectively, these attack scenarios provide a comprehensive evaluation of the proposed framework under both conventional editing operations and modern AI-assisted content manipulation techniques. The aforementioned attacks were evaluated under both single-frame and multi-frame tampering scenarios. In the multi-frame setting, the attack was applied to a randomly selected subset corresponding to 5% of the video frames, allowing the assessment of the proposed

method under more realistic temporal tampering conditions. All experiments were conducted on a Ryzen 7 5700X 8-core CPU with 32 GB RAM running Windows 10.

### 6.7.1 Video: Run-time Efficiency

#### Code-based Hash Chain Video Watermarking Model

The computational efficiency of the proposed model is evaluated by measuring the average run-time per frame at multiple resolutions (720p, 1080p, 1440p, and 4K), frame rates (24, 30, and 60 fps), and video codecs (H.264, HEVC, and AV1).

Table 6.20: Run-time benchmarking across different video resolutions. The time per frame for every resolution is the average of that resolution across all codecs and fps. The benchmark was conducted using parallelized execution of the embedding and extraction procedures utilizing 4 CPU cores.

| Resolution | Hash EM/video(s) | Total EMT(s)                  | Total EXT(s)                  |
|------------|------------------|-------------------------------|-------------------------------|
| 720p       | 0.050            | $(2.41/F \times TF) + 0.050$  | $(2.18/F \times TF) + 0.050$  |
| 1080p      | 0.051            | $(4.21/F \times TF) + 0.051$  | $(3.58/F \times TF) + 0.051$  |
| 2K         | 0.050            | $(7.10/F \times TF) + 0.050$  | $(5.59/F \times TF) + 0.050$  |
| 4K         | 0.053            | $(16.66/F \times TF) + 0.053$ | $(11.65/F \times TF) + 0.053$ |

As shown in Table 6.20, the processing time increases consistently with spatial resolution due to the growing number of pixels, partitions, and block-level frequency operations performed per frame. In the table we denote the Embedding as EM, the Validation as VD, the Embedding Time as EMT, the Extraction Time as EXT, the Frame as F and the Total Frames as TF. Since the embedding procedure applies transform-domain operations within each partition, higher resolutions result in a proportional increase in FFT computations and watermark insertions. The total embedding time per video can be expressed as  $(t_e \cdot TF) + t_h$ , where  $t_e$  denotes the average embedding time per frame,  $TF$  the total number of frames, and  $t_h$  the fixed overhead associated with global hash computation and audio-domain embedding. As observed, the hash-related cost remains approximately constant ( $\approx 0.05$  s per video), while the dominant contribution stems from the linear per-frame embedding component. Similarly, the total extraction can be expressed as  $(t_v \cdot TF) + t_h$ , where  $t_v$



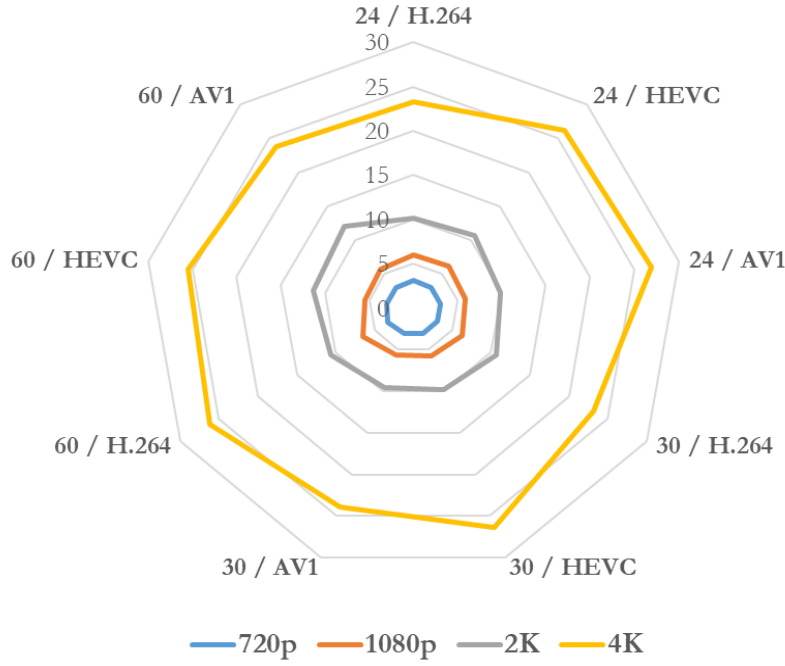


Figure 6.8: Demonstration of the stable scalability and predictability in performance using the average per-frame run-time over different resolutions (720p, 1080p, 2K, and 4K) with frames-rates: 24, 30, 60 FPS, and H.264, HEVC and AV1 codecs for the code-based hash-chain video watermarking model.

denotes the average validation time per frame. Since extraction is blind and processes all partitions regardless of watermark correctness, its computational behavior scales linearly with frame count, while remaining lower than embedding due to the absence of modification operations.

Figure 6.8 further illustrates that resolution is the dominant factor affecting computational cost, whereas variations in frame rate and codec introduce relatively minor differences. This indicates that the proposed method is largely independent of encoding format and temporal sampling, ensuring predictable performance across heterogeneous video sources without requiring codec-specific optimization.

To further improve computational efficiency, a parallel implementation of the proposed model was also developed utilizing Python multiprocessing. Since the watermark embedding procedure operates on independent frame blocks, the workload can be naturally distributed across multiple processing cores, enabling concurrent execution of the embedding and extraction operations. As demonstrated in Table 6.24, the parallel implementation significantly reduces the processing time across all evaluated video resolutions, with the achieved speedup increasing as the video resolution

Table 6.21: Average Embedding Time (ET) in seconds of the serialized and parallel implementations of the code-based hash-chain video watermarking model with corresponding approximate throughput in frames per second (FPS), and the achieved time reduction for different video resolutions. Runtime values are averaged across all tested frame rates and codecs at each resolution using a 4-core CPU. FPS is computed as  $\text{FPS} = 1/\text{Time}$ .

| Resolution | Serialized ET/Frame (s)     | Parallel ET/Frame (s)       | Time Reduction (%) |
|------------|-----------------------------|-----------------------------|--------------------|
| 720p       | 6.53 ( $\approx 0.15$ FPS)  | 2.41 ( $\approx 0.41$ FPS)  | 63.1               |
| 1080p      | 14.56 ( $\approx 0.07$ FPS) | 4.21 ( $\approx 0.24$ FPS)  | 71.1               |
| 2K         | 26.90 ( $\approx 0.04$ FPS) | 7.10 ( $\approx 0.14$ FPS)  | 73.6               |
| 4K         | 64.88 ( $\approx 0.02$ FPS) | 16.66 ( $\approx 0.06$ FPS) | 74.3               |

grows. This behavior indicates that the proposed model exhibits good scalability and can effectively benefit from additional computational resources when processing high-resolution video content.

### Cross-Frame Permutation Based Video Watermarking Model

Table 6.22 summarizes the embedding time required for each video, along with the corresponding permutation size, which reflects the number of elements involved in the cross-frame permutation process. As observed, the embedding time varies across videos, ranging from 13.26 seconds to 99.60 seconds, with an average of 56.33 seconds. This variation is primarily attributed to differences in permutation size, which depends on the total duration of the video. It is worth noting that, unlike the corresponding image-based integrity-preservation model where the permutation is established solely among the *integrity-blocks* of a single image, the proposed video-based framework extends the permutation mapping across both frames and *integrity-blocks*.

More specifically, the permutation is first established among the video frames, and subsequently among the *integrity-blocks* contained within the selected frames. Consequently, the integrity information generated from a specific *integrity-block* may be embedded into an *integrity-block* belonging to a different frame, introducing cross-frame verification dependencies that further strengthen the integrity-preservation ca-

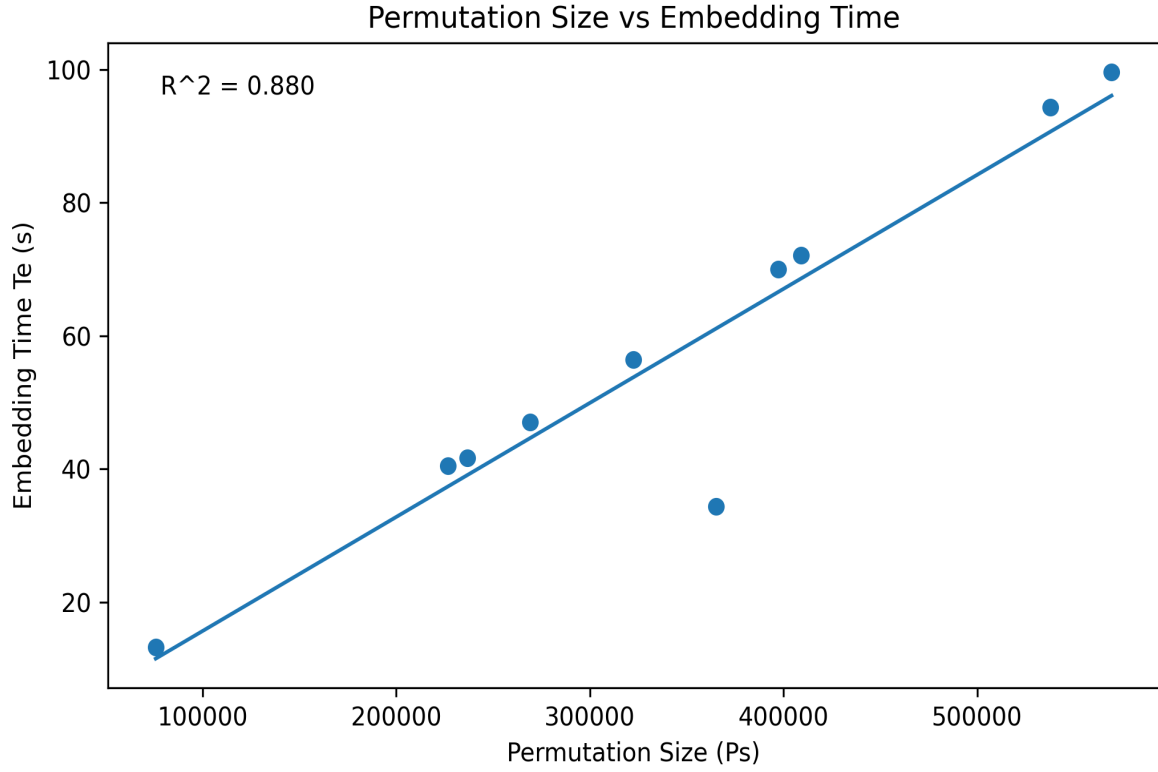


Figure 6.9: Relationship between permutation size ( $P_s$ ) and embedding time ( $T_e$ ) for the evaluated CCTV videos. The observed positive linear trend indicates that the computational cost of the proposed framework increases proportionally to the number of elements participating in the cross-frame permutation process.

pabilities of the proposed framework. Larger permutation sizes result in increased computational overhead due to the higher number of block-level operations and hash computations required during embedding. Specifically, videos with larger permutation sizes (e.g., cctv2 and cctv4) exhibit higher embedding times, while smaller videos (e.g., cctv1) require significantly less processing time. This demonstrates a direct relationship between permutation size and computational cost, confirming the scalability of the proposed method. The results depicted in Figure 6.9 reveal a strong positive linear relationship between the permutation size ( $P_s$ ) and the corresponding embedding time ( $T_e$ ). The high  $R^2$  value of the fitted regression model indicates that the computational cost increases proportionally to the number of elements participating in the cross-frame permutation process. This behavior demonstrates both the scalability and the predictability of the proposed framework, since larger video objects result in increased processing times following a consistent and approximately linear

Table 6.22: Performance evaluation of the proposed cross-frame permutation-based video watermarking model in terms of perceptual quality and computational cost.

| <b>Video</b>   | <b>Duration (s)</b> | <b>PSNR (dB)</b> | <b>SSIM</b>  | $T_e$ (s)    | $P_s$ (elements) |
|----------------|---------------------|------------------|--------------|--------------|------------------|
| cctv1          | 8                   | 42.89            | 0.98         | 13.26        | 75900            |
| cctv2          | 59                  | 42.30            | 0.98         | 94.38        | 537900           |
| cctv3          | 35                  | 50.57            | 0.99         | 56.45        | 322500           |
| cctv4          | 61.8                | 48.28            | 0.99         | 99.60        | 569400           |
| cctv5          | 45                  | 41.23            | 0.98         | 72.14        | 409200           |
| cctv6          | 40                  | 44.13            | 0.98         | 34.34        | 365100           |
| cctv7          | 25                  | 44.26            | 0.99         | 40.46        | 226800           |
| cctv8          | 30                  | 49.43            | 0.99         | 47.03        | 269100           |
| cctv9          | 44                  | 49.24            | 0.99         | 69.99        | 397200           |
| cctv10         | 26                  | 48.81            | 0.99         | 41.69        | 236700           |
| <b>Average</b> | <b>37.38</b>        | <b>46.11</b>     | <b>0.987</b> | <b>56.33</b> | <b>340980</b>    |

trend. Consequently, the obtained results verify that the proposed model remains computationally efficient while enabling reliable estimation of its execution time for larger video data-sets.

## 6.7.2 Video: Quality and Imperceptibility

### Code-based Hash Chain Video Watermarking Model

Table 6.23 presents the PSNR and SSIM results of the proposed video watermarking model across different resolutions, codecs, and frame rates. The obtained results demonstrate high visual quality and imperceptibility under all evaluated configurations. More specifically, the model consistently achieves SSIM values equal to 1.0, indicating that the embedding process introduces negligible structural distortion to the video content. The PSNR measurements further confirm the transparency of the proposed approach, with average values ranging from 32.31 dB to 38.73 dB across the evaluated resolutions. In particular, higher-resolution videos achieve improved

Table 6.23: PSNR and SSIM results of the code-based hash-chain video watermarking model across resolutions, codecs, and frame rates.

| Res.  | Codec | 24 FPS |      | 30 FPS |      | 60 FPS |      | Codec Avg. |      | Res. Avg.                |                          |
|-------|-------|--------|------|--------|------|--------|------|------------|------|--------------------------|--------------------------|
|       |       | PSNR   | SSIM | PSNR   | SSIM | PSNR   | SSIM | PSNR       | SSIM | $\overline{\text{PSNR}}$ | $\overline{\text{SSIM}}$ |
| 720p  | AV1   | 34.55  | 1.0  | 34.48  | 1.0  | 34.56  | 1.0  | 34.53      | 1.0  | 34.51                    | 1.0                      |
|       | H.264 | 34.40  | 1.0  | 34.55  | 1.0  | 34.57  | 1.0  | 34.51      | 1.0  |                          |                          |
|       | HEVC  | 34.55  | 1.0  | 34.32  | 1.0  | 34.64  | 1.0  | 34.50      | 1.0  |                          |                          |
| 1080p | AV1   | 32.24  | 1.0  | 31.38  | 1.0  | 33.29  | 1.0  | 32.30      | 1.0  | 32.31                    | 1.0                      |
|       | H.264 | 32.18  | 1.0  | 31.37  | 1.0  | 33.43  | 1.0  | 32.33      | 1.0  |                          |                          |
|       | HEVC  | 32.34  | 1.0  | 31.30  | 1.0  | 33.22  | 1.0  | 32.29      | 1.0  |                          |                          |
| 2K    | AV1   | 39.91  | 1.0  | 35.80  | 1.0  | 40.47  | 1.0  | 38.73      | 1.0  | 38.73                    | 1.0                      |
|       | H.264 | 39.75  | 1.0  | 35.80  | 1.0  | 40.55  | 1.0  | 38.70      | 1.0  |                          |                          |
|       | HEVC  | 39.90  | 1.0  | 35.76  | 1.0  | 40.60  | 1.0  | 38.75      | 1.0  |                          |                          |
| 4K    | AV1   | 35.26  | 1.0  | 35.04  | 1.0  | 33.55  | 1.0  | 34.62      | 1.0  | 34.59                    | 1.0                      |
|       | H.264 | 35.04  | 1.0  | 34.87  | 1.0  | 33.58  | 1.0  | 34.50      | 1.0  |                          |                          |
|       | HEVC  | 35.10  | 1.0  | 35.26  | 1.0  | 33.60  | 1.0  | 34.65      | 1.0  |                          |                          |

PSNR values due to their increased spatial capacity for embedding. Furthermore, the limited variation observed across different codecs (AV1, H.264, and HEVC) and frame rates (24, 30, and 60 FPS) indicates that the visual impact of the watermark remains stable regardless of the encoding configuration. The embedding payload of the proposed model is relatively low and is directly determined by the size of the deployed cryptographic hash function. In the proposed implementation, SHA-256 is utilized, resulting in a payload of only 256 bits. More specifically, the hash value computed over the entire set of video frames is embedded into 256 selected *audio-samples* of the audio stream. Consequently, only a very small portion of the available audio samples

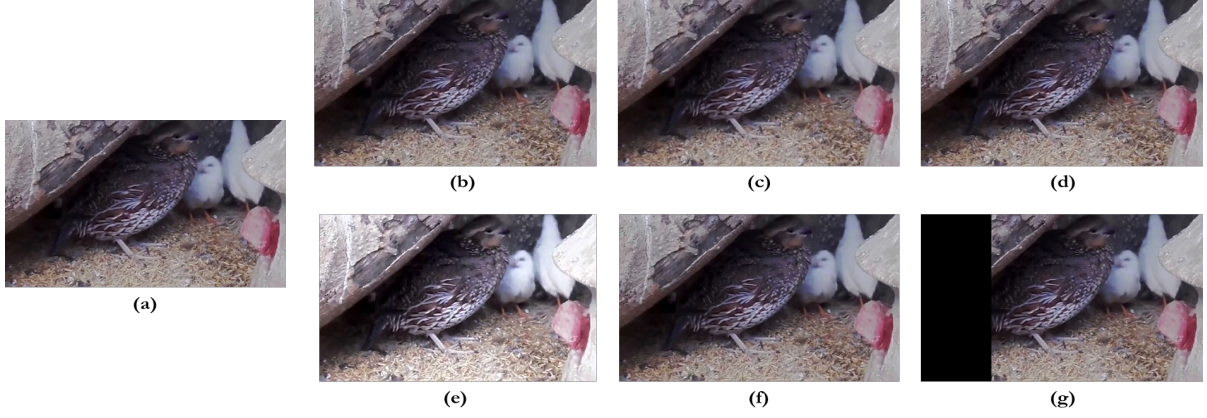


Figure 6.10: Visual quality comparison of the code-based hash chain video watermarking model between the original frame (a), the watermarked frame (b), and frames degraded by common signal processing and geometric attacks, including Gaussian blur ( $5 \times 5$ ) (c), Gaussian noise ( $\sigma = 0.4$ ) (d), histogram equalization (e), JPEG compression ( $Q=50$ ) (f), and left-side cropping (50%) (g).

is modified, introducing negligible distortion to the audio signal.

Overall, the obtained results demonstrate that the proposed model preserves high visual fidelity while maintaining the embedding capacity required for authenticity preservation and tamper detection. In Figure 6.10 it is illustrated a visual example for subjective quality evaluation of the proposed technique considering the embedding of the watermark in a single frame of a video of 4K resolution, alongside the same frame after the performance of various types of attacks, including blurring, cropping, enhance and noise.

### Cross-Frame Permutation Based Video Watermarking Model

As presented in Table 6.22, the proposed framework achieves PSNR values ranging from 41.23 dB to 50.57 dB, with an average value of 46.11 dB. These measurements indicate that the embedding process introduces only limited visual distortion, preserving the perceptual quality of the protected video content. Since PSNR values above 30 dB are generally considered indicative of high-quality reconstruction, the obtained results confirm the imperceptibility of the proposed watermarking strategy. Similarly, the achieved SSIM values range between 0.98 and 0.99, with an average value of 0.987. Such measurements indicate an almost identical structural representation between the

original and watermarked video frames, demonstrating that the proposed framework preserves the visual characteristics and spatial information of the watermarked CCTV videos.

The payload embedded into the video stream is directly related to the number of *integrity-blocks* and video frames participating in the embedding process. Since each *integrity-block* stores a 256-bit SHA-256 hash value, the total payload can be calculated as  $256 \times TotalFrameBlocks \times TotalFrames$ . In the proposed implementation, each frame is partitioned into 300 *integrity-blocks*, resulting in  $256 \times 300 \times TotalFrames$  or equivalently,  $76,800 \times TotalFrames$ , where *TotalFrames* denotes the total number of frames contained in the input video. Consequently, the total payload increases linearly with the video duration, while remaining proportional to the number of *integrity-blocks* participating in the cross-frame permutation and verification process. Despite the relatively high payload introduced by the proposed framework, the modifications are distributed across the LSBs of the video frames, resulting in PSNR and SSIM measurements that confirm the preservation of the visual quality of the protected video content.

Overall, the obtained PSNR and SSIM measurements verify that the proposed cross-frame permutation-based watermarking framework maintains high visual fidelity while embedding authenticity and integrity information, making it suitable for surveillance and forensic video applications where perceptual quality preservation is of critical importance.

### 6.7.3 Video: Authenticity Verification

#### Code-based Hash Chain Video Watermarking Model

Table 6.24 presents the robustness and tamper detection results obtained for 720p videos encoded using H.264, HEVC, and AV1 at 24, 30, and 60 FPS under the examined attack scenarios. Overall, the proposed model demonstrates strong robustness across all codecs and frame rates, with extraction rates ( $R_e$ ) consistently remaining close to 100% and BER values remaining very low in the majority of cases. Furthermore, the relatively small standard deviations indicate that the performance of the model is largely unaffected by variations in frame rate. The highest robustness is observed under Gaussian noise attacks ( $\sigma = 0.4$  and  $\sigma = 0.5$ ), histogram equalization, mild gamma correction ( $\gamma = 1.25$ ), and horizontal cropping of 25%, where average

extraction rates exceed 99% while BER values remain close to 0.001. In particular, HEVC achieves the highest extraction performance under several attack scenarios, with extraction rates approaching 100%.

Performance degradation becomes more noticeable under severe spatial distortions, including large-kernel Gaussian blurring ( $9 \times 9$ ), aggressive JPEG compression ( $Q = 70$ ), and strong gamma correction ( $\gamma = 2.5$ ). Nevertheless, even under these challenging conditions, extraction rates remain above 89% while BER values remain below 0.05, demonstrating the resilience of the proposed watermarking model. A comparison among codecs reveals similar robustness characteristics for H.264 and HEVC, with HEVC exhibiting slightly improved performance in certain noise and cropping scenarios. AV1 generally produces marginally lower extraction rates; however, its average performance remains above 90% across all evaluated attacks. The obtained results demonstrate that the proposed model maintains reliable watermark extraction across different codecs, frame rates, and attack categories, exhibiting only moderate performance degradation even under severe processing conditions. Finally, for all evaluated attack scenarios, the proposed model achieved a 100% tamper detection rate with Accuracy, Precision, Recall, and F1-score metrics consistently approached 100%, indicating that the model is capable of reliably identifying tampered video content while maintaining an extremely low false-positive and false-negative rate.

Table 6.25 presents the robustness results obtained for 1080p videos encoded using H.264, HEVC, and AV1 at 24, 30, and 60 FPS. The proposed model maintains high extraction rates and low BER values across all codecs, frame rates, and attack scenarios, while the small standard deviations indicate limited sensitivity to frame-rate variations. The highest robustness is observed under Gaussian noise, light filtering, histogram equalization, moderate gamma correction ( $\gamma = 1.25$ ), and horizontal cropping, where extraction rates typically exceed 99%. The most challenging attacks are large-kernel Gaussian blur, aggressive compression ( $Q = 70$ ), and strong gamma correction ( $\gamma = 2.5$ ), which reduce extraction rates to approximately 89% – 93%. Nevertheless, watermark recovery remains successful in all cases. H.264 and HEVC exhibit comparable robustness, with HEVC providing slightly improved performance under severe distortions, while AV1 demonstrates marginally lower extraction rates. The exhibited results follow the trends observed for the 720p experiments, confirming the scalability of the proposed model to higher resolutions. Furthermore, all attack scenarios achieved a 100% tamper detection rate, with Accuracy, Precision, Recall,



Table 6.24: Verification and robustness results for the code-based hash-chain video watermarking model on 720p videos of H264, HEVC, and AV1 codecs across 24/30/60 FPS, depicting the mean  $\pm$  standard deviation Extraction Rate( $R_e$ ) and BER for each row.

|                           |                           | 24 FPS |       |       | 30 FPS |       |       | 60 FPS |       |              | Average                           |                                   |
|---------------------------|---------------------------|--------|-------|-------|--------|-------|-------|--------|-------|--------------|-----------------------------------|-----------------------------------|
| Attack                    |                           | $R_e$  | BER   | T     | $R_e$  | BER   | T     | $R_e$  | BER   | T            | $\overline{R_e} \pm \sigma_{R_e}$ | $\overline{BER} \pm \sigma_{BER}$ |
| H264                      | No Attack                 | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗            | 100.0 ± 0.00                      | 0.000 ± 0.000                     |
|                           | Blur Average (3 × 3)      | 99.48  | 0.002 | ✓     | 99.62  | 0.002 | ✓     | 99.61  | 0.001 | ✓            | 99.57 ± 0.07                      | 0.002 ± 0.001                     |
|                           | Blur Average (7 × 7)      | 91.54  | 0.033 | ✓     | 91.58  | 0.037 | ✓     | 91.33  | 0.038 | ✓            | 91.48 ± 0.11                      | 0.036 ± 0.002                     |
|                           | Blur Gaussian (5 × 5)     | 99.38  | 0.003 | ✓     | 99.56  | 0.002 | ✓     | 99.60  | 0.001 | ✓            | 99.51 ± 0.09                      | 0.002 ± 0.001                     |
|                           | Blur Gaussian (9 × 9)     | 90.99  | 0.042 | ✓     | 91.15  | 0.043 | ✓     | 91.59  | 0.043 | ✓            | 91.24 ± 0.25                      | 0.043 ± 0.001                     |
|                           | Blur Median(3 × 3)        | 92.68  | 0.029 | ✓     | 93.02  | 0.029 | ✓     | 94.04  | 0.029 | ✓            | 93.25 ± 0.57                      | 0.029 ± 0.000                     |
|                           | Blur Median(9 × 9)        | 99.38  | 0.002 | ✓     | 99.56  | 0.002 | ✓     | 99.61  | 0.001 | ✓            | 99.52 ± 0.12                      | 0.002 ± 0.001                     |
|                           | Compression (Q=50)        | 99.45  | 0.002 | ✓     | 99.60  | 0.002 | ✓     | 99.61  | 0.001 | ✓            | 99.55 ± 0.08                      | 0.002 ± 0.001                     |
|                           | Compression (Q=70)        | 93.18  | 0.025 | ✓     | 93.40  | 0.028 | ✓     | 93.18  | 0.028 | ✓            | 93.25 ± 0.13                      | 0.027 ± 0.002                     |
|                           | Noise Gaussian (σ=0.4)    | 99.77  | 0.001 | ✓     | 99.88  | 0.001 | ✓     | 99.74  | 0.001 | ✓            | 99.80 ± 0.06                      | 0.001 ± 0.000                     |
|                           | Noise Gaussian (σ=0.5)    | 99.77  | 0.001 | ✓     | 99.88  | 0.001 | ✓     | 99.74  | 0.001 | ✓            | 99.80 ± 0.06                      | 0.001 ± 0.000                     |
|                           | Salt and Pepper           | 99.40  | 0.002 | ✓     | 99.54  | 0.002 | ✓     | 99.60  | 0.001 | ✓            | 99.51 ± 0.10                      | 0.002 ± 0.001                     |
|                           | Crop Height (top, 50%)    | 94.79  | 0.020 | ✓     | 95.52  | 0.020 | ✓     | 95.40  | 0.020 | ✓            | 95.24 ± 0.31                      | 0.020 ± 0.000                     |
|                           | Crop Width (left, 25%)    | 99.69  | 0.001 | ✓     | 99.81  | 0.001 | ✓     | 99.71  | 0.001 | ✓            | 99.74 ± 0.06                      | 0.001 ± 0.000                     |
| HEVC                      | Gamma Correction (γ=1.25) | 99.51  | 0.002 | ✓     | 99.62  | 0.001 | ✓     | 99.64  | 0.001 | ✓            | 99.59 ± 0.06                      | 0.001 ± 0.001                     |
|                           | Gamma Correction (γ=2.5)  | 93.67  | 0.025 | ✓     | 92.69  | 0.026 | ✓     | 94.16  | 0.025 | ✓            | 93.51 ± 0.75                      | 0.025 ± 0.001                     |
|                           | Histogram Equalization    | 99.74  | 0.001 | ✓     | 99.88  | 0.001 | ✓     | 99.74  | 0.001 | ✓            | 99.79 ± 0.08                      | 0.001 ± 0.000                     |
|                           | No Attack                 | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗            | 100.0 ± 0.00                      | 0.000 ± 0.000                     |
|                           | Blur Average (3 × 3)      | 99.66  | 0.002 | ✓     | 99.67  | 0.002 | ✓     | 99.53  | 0.002 | ✓            | 99.62 ± 0.06                      | 0.002 ± 0.000                     |
|                           | Blur Average (7 × 7)      | 91.59  | 0.035 | ✓     | 91.62  | 0.036 | ✓     | 91.99  | 0.037 | ✓            | 91.73 ± 0.20                      | 0.036 ± 0.001                     |
|                           | Blur Gaussian (5 × 5)     | 99.64  | 0.003 | ✓     | 99.58  | 0.002 | ✓     | 99.40  | 0.002 | ✓            | 99.54 ± 0.12                      | 0.002 ± 0.001                     |
|                           | Blur Gaussian (9 × 9)     | 92.24  | 0.043 | ✓     | 91.54  | 0.041 | ✓     | 91.62  | 0.046 | ✓            | 91.80 ± 0.36                      | 0.043 ± 0.002                     |
|                           | Blur Median(3 × 3)        | 92.63  | 0.030 | ✓     | 93.12  | 0.030 | ✓     | 93.41  | 0.031 | ✓            | 93.05 ± 0.39                      | 0.030 ± 0.001                     |
|                           | Blur Median(9 × 9)        | 99.61  | 0.002 | ✓     | 99.60  | 0.002 | ✓     | 99.40  | 0.002 | ✓            | 99.54 ± 0.11                      | 0.002 ± 0.000                     |
|                           | Compression (Q=50)        | 99.64  | 0.002 | ✓     | 99.65  | 0.002 | ✓     | 99.43  | 0.002 | ✓            | 99.57 ± 0.12                      | 0.002 ± 0.000                     |
|                           | Compression (Q=70)        | 92.45  | 0.027 | ✓     | 92.73  | 0.027 | ✓     | 94.51  | 0.029 | ✓            | 93.23 ± 1.06                      | 0.028 ± 0.001                     |
|                           | Noise Gaussian (σ=0.4)    | 99.97  | 0.001 | ✓     | 99.90  | 0.001 | ✓     | 99.53  | 0.002 | ✓            | 99.80 ± 0.23                      | 0.001 ± 0.001                     |
|                           | Noise Gaussian (σ=0.5)    | 99.97  | 0.001 | ✓     | 99.90  | 0.001 | ✓     | 99.53  | 0.002 | ✓            | 99.80 ± 0.23                      | 0.001 ± 0.001                     |
| AV1                       | Salt and Pepper           | 99.61  | 0.002 | ✓     | 99.60  | 0.002 | ✓     | 99.38  | 0.002 | ✓            | 99.53 ± 0.12                      | 0.002 ± 0.000                     |
|                           | Crop Height (top, 50%)    | 96.22  | 0.020 | ✓     | 94.90  | 0.020 | ✓     | 95.17  | 0.021 | ✓            | 95.43 ± 0.67                      | 0.020 ± 0.001                     |
|                           | Crop Width (left, 25%)    | 99.87  | 0.001 | ✓     | 99.81  | 0.001 | ✓     | 99.51  | 0.002 | ✓            | 99.73 ± 0.18                      | 0.001 ± 0.001                     |
|                           | Gamma Correction (γ=1.25) | 99.71  | 0.002 | ✓     | 99.69  | 0.001 | ✓     | 99.45  | 0.002 | ✓            | 99.62 ± 0.13                      | 0.002 ± 0.001                     |
|                           | Gamma Correction (γ=2.5)  | 93.49  | 0.025 | ✓     | 93.58  | 0.025 | ✓     | 93.59  | 0.027 | ✓            | 93.55 ± 0.06                      | 0.026 ± 0.001                     |
|                           | Histogram Equalization    | 99.74  | 0.002 | ✓     | 99.90  | 0.001 | ✓     | 99.53  | 0.002 | ✓            | 99.72 ± 0.19                      | 0.002 ± 0.001                     |
|                           | No Attack                 | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗            | 100.0 ± 0.00                      | 0.000 ± 0.000                     |
|                           | Blur Average (3 × 3)      | 98.80  | 0.002 | ✓     | 98.94  | 0.002 | ✓     | 99.55  | 0.001 | ✓            | 99.10 ± 0.37                      | 0.002 ± 0.001                     |
|                           | Blur Average (7 × 7)      | 90.57  | 0.034 | ✓     | 90.50  | 0.038 | ✓     | 90.39  | 0.034 | ✓            | 90.49 ± 0.09                      | 0.035 ± 0.002                     |
|                           | Blur Gaussian (5 × 5)     | 98.75  | 0.003 | ✓     | 98.83  | 0.002 | ✓     | 99.50  | 0.001 | ✓            | 99.03 ± 0.38                      | 0.002 ± 0.001                     |
|                           | Blur Gaussian (9 × 9)     | 90.49  | 0.042 | ✓     | 89.81  | 0.041 | ✓     | 89.76  | 0.039 | ✓            | 90.02 ± 0.37                      | 0.041 ± 0.002                     |
|                           | Blur Median(3 × 3)        | 92.40  | 0.030 | ✓     | 92.29  | 0.030 | ✓     | 91.95  | 0.029 | ✓            | 92.21 ± 0.23                      | 0.030 ± 0.001                     |
|                           | Blur Median(9 × 9)        | 98.75  | 0.003 | ✓     | 98.88  | 0.002 | ✓     | 99.50  | 0.001 | ✓            | 99.04 ± 0.37                      | 0.002 ± 0.001                     |
|                           | Compression (Q=50)        | 98.83  | 0.002 | ✓     | 98.94  | 0.002 | ✓     | 99.53  | 0.001 | ✓            | 99.10 ± 0.36                      | 0.002 ± 0.001                     |
| Compression (Q=70)        | 92.01                     | 0.027  | ✓     | 92.46 | 0.027  | ✓     | 92.72 | 0.026  | ✓     | 92.40 ± 0.36 | 0.027 ± 0.001                     |                                   |
| Noise Gaussian (σ=0.4)    | 99.14                     | 0.001  | ✓     | 99.17 | 0.001  | ✓     | 99.67 | 0.001  | ✓     | 99.33 ± 0.29 | 0.001 ± 0.000                     |                                   |
| Noise Gaussian (σ=0.5)    | 99.14                     | 0.001  | ✓     | 99.17 | 0.001  | ✓     | 99.67 | 0.001  | ✓     | 99.33 ± 0.29 | 0.001 ± 0.000                     |                                   |
| Salt and Pepper           | 98.72                     | 0.002  | ✓     | 98.88 | 0.002  | ✓     | 99.52 | 0.001  | ✓     | 99.04 ± 0.42 | 0.002 ± 0.001                     |                                   |
| Crop Height (top, 50%)    | 94.22                     | 0.020  | ✓     | 94.25 | 0.020  | ✓     | 94.69 | 0.020  | ✓     | 94.39 ± 0.21 | 0.020 ± 0.000                     |                                   |
| Crop Width (left, 25%)    | 99.04                     | 0.001  | ✓     | 99.08 | 0.001  | ✓     | 99.62 | 0.001  | ✓     | 99.25 ± 0.30 | 0.001 ± 0.000                     |                                   |
| Gamma Correction (γ=1.25) | 98.91                     | 0.002  | ✓     | 98.94 | 0.002  | ✓     | 99.54 | 0.001  | ✓     | 99.13 ± 0.33 | 0.002 ± 0.001                     |                                   |
| Gamma Correction (γ=2.5)  | 92.71                     | 0.026  | ✓     | 92.60 | 0.025  | ✓     | 92.36 | 0.025  | ✓     | 92.56 ± 0.18 | 0.025 ± 0.001                     |                                   |
| Histogram Equalization    | 99.14                     | 0.001  | ✓     | 98.98 | 0.002  | ✓     | 99.67 | 0.001  | ✓     | 99.26 ± 0.35 | 0.001 ± 0.001                     |                                   |

Table 6.25: Verification and robustness results for the code-based hash-chain video watermarking model on 1080p videos of H264, HEVC, and AV1 codecs across 24/30/60 FPS, depicting the mean  $\pm$  standard deviation Extraction Rate( $R_e$ ) and BER for each row.

|                          |                           | 24 FPS |       |       | 30 FPS |       |       | 60 FPS |       |              | Average                           |                                   |
|--------------------------|---------------------------|--------|-------|-------|--------|-------|-------|--------|-------|--------------|-----------------------------------|-----------------------------------|
| Attack                   |                           | $R_e$  | BER   | T     | $R_e$  | BER   | T     | $R_e$  | BER   | T            | $\overline{R_e} \pm \sigma_{R_e}$ | $\overline{BER} \pm \sigma_{BER}$ |
| H264                     | No Attack                 | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗            | 100.0 ± 0.00                      | 0.000 ± 0.000                     |
|                          | Blur Average (3 × 3)      | 99.32  | 0.002 | ✓     | 99.71  | 0.002 | ✓     | 99.42  | 0.002 | ✓            | 99.48 ± 0.20                      | 0.002 ± 0.000                     |
|                          | Blur Average (7 × 7)      | 90.76  | 0.042 | ✓     | 92.44  | 0.035 | ✓     | 91.67  | 0.037 | ✓            | 91.62 ± 0.69                      | 0.038 ± 0.003                     |
|                          | Blur Gaussian (5 × 5)     | 99.58  | 0.002 | ✓     | 95.96  | 0.024 | ✓     | 99.65  | 0.001 | ✓            | 98.40 ± 1.72                      | 0.009 ± 0.011                     |
|                          | Blur Gaussian (9 × 9)     | 94.06  | 0.044 | ✓     | 93.33  | 0.047 | ✓     | 92.44  | 0.043 | ✓            | 93.28 ± 0.66                      | 0.045 ± 0.002                     |
|                          | Blur Median(3 × 3)        | 93.80  | 0.026 | ✓     | 94.69  | 0.025 | ✓     | 91.89  | 0.032 | ✓            | 93.46 ± 1.16                      | 0.028 ± 0.003                     |
|                          | Blur Median(9 × 9)        | 99.95  | 0.000 | ✓     | 99.73  | 0.001 | ✓     | 99.58  | 0.001 | ✓            | 99.75 ± 0.15                      | 0.001 ± 0.000                     |
|                          | Compression (Q=50)        | 99.66  | 0.001 | ✓     | 99.40  | 0.003 | ✓     | 99.51  | 0.001 | ✓            | 99.52 ± 0.13                      | 0.002 ± 0.001                     |
|                          | Compression (Q=70)        | 93.44  | 0.027 | ✓     | 92.58  | 0.035 | ✓     | 92.10  | 0.032 | ✓            | 92.71 ± 0.55                      | 0.031 ± 0.004                     |
|                          | Noise Gaussian (σ=0.4)    | 99.95  | 0.000 | ✓     | 99.90  | 0.001 | ✓     | 99.65  | 0.001 | ✓            | 99.83 ± 0.13                      | 0.001 ± 0.000                     |
|                          | Noise Gaussian (σ=0.5)    | 99.95  | 0.000 | ✓     | 99.90  | 0.001 | ✓     | 99.65  | 0.001 | ✓            | 99.83 ± 0.13                      | 0.001 ± 0.000                     |
|                          | Salt and Pepper           | 96.55  | 0.015 | ✓     | 95.97  | 0.019 | ✓     | 95.42  | 0.019 | ✓            | 95.98 ± 0.46                      | 0.018 ± 0.002                     |
|                          | Crop Height (top, 50%)    | 95.36  | 0.018 | ✓     | 95.83  | 0.019 | ✓     | 95.62  | 0.019 | ✓            | 95.60 ± 0.19                      | 0.019 ± 0.000                     |
|                          | Crop Width (left, 25%)    | 99.84  | 0.000 | ✓     | 99.83  | 0.001 | ✓     | 99.60  | 0.001 | ✓            | 99.76 ± 0.14                      | 0.001 ± 0.000                     |
|                          | Gamma Correction (γ=1.25) | 99.95  | 0.000 | ✓     | 99.88  | 0.001 | ✓     | 99.65  | 0.001 | ✓            | 99.83 ± 0.13                      | 0.001 ± 0.000                     |
| Gamma Correction (γ=2.5) | 99.48                     | 0.001  | ✓     | 99.33 | 0.002  | ✓     | 98.99 | 0.001  | ✓     | 99.27 ± 0.20 | 0.001 ± 0.000                     |                                   |
| Histogram Equalization   | 99.27                     | 0.003  | ✓     | 99.53 | 0.002  | ✓     | 99.24 | 0.002  | ✓     | 99.35 ± 0.16 | 0.002 ± 0.001                     |                                   |
| HEVC                     | No Attack                 | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗            | 100.0 ± 0.00                      | 0.000 ± 0.000                     |
|                          | Blur Average (3 × 3)      | 99.51  | 0.002 | ✓     | 99.73  | 0.002 | ✓     | 99.42  | 0.002 | ✓            | 99.55 ± 0.13                      | 0.002 ± 0.000                     |
|                          | Blur Average (7 × 7)      | 91.02  | 0.038 | ✓     | 92.03  | 0.036 | ✓     | 91.58  | 0.037 | ✓            | 91.54 ± 0.41                      | 0.037 ± 0.001                     |
|                          | Blur Gaussian (5 × 5)     | 99.46  | 0.003 | ✓     | 96.31  | 0.021 | ✓     | 99.41  | 0.002 | ✓            | 98.39 ± 1.46                      | 0.009 ± 0.011                     |
|                          | Blur Gaussian (9 × 9)     | 94.23  | 0.043 | ✓     | 93.51  | 0.042 | ✓     | 92.91  | 0.045 | ✓            | 93.55 ± 0.54                      | 0.043 ± 0.002                     |
|                          | Blur Median(3 × 3)        | 93.68  | 0.028 | ✓     | 94.01  | 0.028 | ✓     | 92.84  | 0.031 | ✓            | 93.51 ± 0.49                      | 0.029 ± 0.002                     |
|                          | Blur Median(9 × 9)        | 99.83  | 0.001 | ✓     | 99.62  | 0.002 | ✓     | 99.45  | 0.002 | ✓            | 99.63 ± 0.16                      | 0.002 ± 0.001                     |
|                          | Compression (Q=50)        | 99.53  | 0.002 | ✓     | 99.41  | 0.002 | ✓     | 99.37  | 0.002 | ✓            | 99.44 ± 0.07                      | 0.002 ± 0.000                     |
|                          | Compression (Q=70)        | 92.88  | 0.030 | ✓     | 92.64  | 0.029 | ✓     | 93.51  | 0.028 | ✓            | 93.01 ± 0.36                      | 0.029 ± 0.001                     |
|                          | Noise Gaussian (σ=0.4)    | 99.92  | 0.001 | ✓     | 99.88  | 0.001 | ✓     | 99.61  | 0.002 | ✓            | 99.80 ± 0.14                      | 0.001 ± 0.001                     |
|                          | Noise Gaussian (σ=0.5)    | 99.92  | 0.001 | ✓     | 99.88  | 0.001 | ✓     | 99.61  | 0.002 | ✓            | 99.80 ± 0.14                      | 0.001 ± 0.001                     |
|                          | Salt and Pepper           | 96.12  | 0.017 | ✓     | 95.76  | 0.019 | ✓     | 95.33  | 0.020 | ✓            | 95.74 ± 0.32                      | 0.019 ± 0.002                     |
|                          | Crop Height (top, 50%)    | 96.04  | 0.018 | ✓     | 95.61  | 0.019 | ✓     | 95.29  | 0.020 | ✓            | 95.65 ± 0.31                      | 0.019 ± 0.001                     |
|                          | Crop Width (left, 25%)    | 99.82  | 0.001 | ✓     | 99.79  | 0.001 | ✓     | 99.52  | 0.002 | ✓            | 99.71 ± 0.16                      | 0.001 ± 0.001                     |
|                          | Gamma Correction (γ=1.25) | 99.88  | 0.001 | ✓     | 99.83  | 0.001 | ✓     | 99.58  | 0.002 | ✓            | 99.76 ± 0.16                      | 0.001 ± 0.001                     |
| Gamma Correction (γ=2.5) | 99.26                     | 0.002  | ✓     | 99.11 | 0.002  | ✓     | 98.88 | 0.002  | ✓     | 99.08 ± 0.16 | 0.002 ± 0.000                     |                                   |
| Histogram Equalization   | 99.41                     | 0.002  | ✓     | 99.52 | 0.002  | ✓     | 99.33 | 0.002  | ✓     | 99.42 ± 0.10 | 0.002 ± 0.000                     |                                   |
| AV1                      | No Attack                 | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗            | 100.0 ± 0.00                      | 0.000 ± 0.000                     |
|                          | Blur Average (3 × 3)      | 98.94  | 0.002 | ✓     | 99.12  | 0.002 | ✓     | 99.51  | 0.001 | ✓            | 99.19 ± 0.29                      | 0.002 ± 0.001                     |
|                          | Blur Average (7 × 7)      | 90.23  | 0.036 | ✓     | 90.61  | 0.035 | ✓     | 90.44  | 0.034 | ✓            | 90.43 ± 0.16                      | 0.035 ± 0.001                     |
|                          | Blur Gaussian (5 × 5)     | 98.71  | 0.003 | ✓     | 98.93  | 0.002 | ✓     | 99.46  | 0.001 | ✓            | 99.03 ± 0.38                      | 0.002 ± 0.001                     |
|                          | Blur Gaussian (9 × 9)     | 90.32  | 0.042 | ✓     | 89.77  | 0.041 | ✓     | 89.68  | 0.039 | ✓            | 89.92 ± 0.28                      | 0.041 ± 0.002                     |
|                          | Blur Median(3 × 3)        | 92.17  | 0.030 | ✓     | 92.33  | 0.030 | ✓     | 91.82  | 0.029 | ✓            | 92.11 ± 0.22                      | 0.030 ± 0.001                     |
|                          | Blur Median(9 × 9)        | 98.69  | 0.003 | ✓     | 98.90  | 0.002 | ✓     | 99.43  | 0.001 | ✓            | 99.01 ± 0.38                      | 0.002 ± 0.001                     |
|                          | Compression (Q=50)        | 98.81  | 0.002 | ✓     | 98.96  | 0.002 | ✓     | 99.47  | 0.001 | ✓            | 99.08 ± 0.34                      | 0.002 ± 0.001                     |
|                          | Compression (Q=70)        | 91.94  | 0.027 | ✓     | 92.37  | 0.027 | ✓     | 92.64  | 0.026 | ✓            | 92.32 ± 0.29                      | 0.027 ± 0.001                     |
|                          | Noise Gaussian (σ=0.4)    | 99.09  | 0.001 | ✓     | 99.22  | 0.001 | ✓     | 99.63  | 0.001 | ✓            | 99.31 ± 0.28                      | 0.001 ± 0.000                     |
|                          | Noise Gaussian (σ=0.5)    | 99.09  | 0.001 | ✓     | 99.22  | 0.001 | ✓     | 99.63  | 0.001 | ✓            | 99.31 ± 0.28                      | 0.001 ± 0.000                     |
|                          | Salt and Pepper           | 98.64  | 0.002 | ✓     | 98.82  | 0.002 | ✓     | 99.44  | 0.001 | ✓            | 98.97 ± 0.42                      | 0.002 ± 0.001                     |
|                          | Crop Height (top, 50%)    | 94.14  | 0.020 | ✓     | 94.32  | 0.020 | ✓     | 94.61  | 0.020 | ✓            | 94.36 ± 0.20                      | 0.020 ± 0.000                     |
|                          | Crop Width (left, 25%)    | 98.97  | 0.001 | ✓     | 99.14  | 0.001 | ✓     | 99.58  | 0.001 | ✓            | 99.23 ± 0.31                      | 0.001 ± 0.000                     |
|                          | Gamma Correction (γ=1.25) | 98.84  | 0.002 | ✓     | 98.97  | 0.002 | ✓     | 99.45  | 0.001 | ✓            | 99.09 ± 0.33                      | 0.002 ± 0.001                     |
| Gamma Correction (γ=2.5) | 92.58                     | 0.026  | ✓     | 92.71 | 0.025  | ✓     | 92.33 | 0.025  | ✓     | 92.54 ± 0.19 | 0.025 ± 0.001                     |                                   |
| Histogram Equalization   | 99.08                     | 0.001  | ✓     | 99.02 | 0.002  | ✓     | 99.63 | 0.001  | ✓     | 99.24 ± 0.34 | 0.001 ± 0.001                     |                                   |

Table 6.26: Verification and robustness results for the code-based hash-chain video watermarking model on 2K videos of H264, HEVC, and AV1 codecs across 24/30/60 FPS, depicting the mean  $\pm$  standard deviation Extraction Rate( $R_e$ ) and BER for each row.

|                                    |                                    | 24 FPS |       |       | 30 FPS |       |       | 60 FPS |       |              | Average                           |                                   |
|------------------------------------|------------------------------------|--------|-------|-------|--------|-------|-------|--------|-------|--------------|-----------------------------------|-----------------------------------|
| Attack                             |                                    | $R_e$  | BER   | T     | $R_e$  | BER   | T     | $R_e$  | BER   | T            | $\overline{R_e} \pm \sigma_{R_e}$ | $\overline{BER} \pm \sigma_{BER}$ |
| H264                               | No Attack                          | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗            | 100.0 ± 0.00                      | 0.000 ± 0.000                     |
|                                    | Blur Average (3 × 3)               | 99.77  | 0.000 | ✓     | 99.79  | 0.000 | ✓     | 99.82  | 0.000 | ✓            | 99.79 ± 0.02                      | 0.000 ± 0.000                     |
|                                    | Blur Average (7 × 7)               | 94.11  | 0.032 | ✓     | 93.33  | 0.036 | ✓     | 94.02  | 0.030 | ✓            | 93.82 ± 0.34                      | 0.033 ± 0.003                     |
|                                    | Blur Gaussian (5 × 5)              | 99.48  | 0.002 | ✓     | 95.14  | 0.023 | ✓     | 99.65  | 0.001 | ✓            | 98.09 ± 2.03                      | 0.009 ± 0.010                     |
|                                    | Blur Gaussian (9 × 9)              | 91.07  | 0.050 | ✓     | 92.56  | 0.042 | ✓     | 90.27  | 0.045 | ✓            | 91.30 ± 0.94                      | 0.046 ± 0.003                     |
|                                    | Blur Median (3 × 3)                | 95.83  | 0.011 | ✓     | 94.00  | 0.009 | ✓     | 93.77  | 0.011 | ✓            | 94.53 ± 0.92                      | 0.010 ± 0.001                     |
|                                    | Blur Median (9 × 9)                | 99.40  | 0.001 | ✓     | 99.88  | 0.000 | ✓     | 99.80  | 0.000 | ✓            | 99.69 ± 0.21                      | 0.000 ± 0.000                     |
|                                    | Compression (Q=50)                 | 99.53  | 0.001 | ✓     | 99.79  | 0.000 | ✓     | 99.83  | 0.000 | ✓            | 99.72 ± 0.16                      | 0.000 ± 0.000                     |
|                                    | Compression (Q=70)                 | 96.43  | 0.009 | ✓     | 95.10  | 0.008 | ✓     | 94.70  | 0.010 | ✓            | 95.41 ± 0.73                      | 0.009 ± 0.001                     |
|                                    | Noise Gaussian ( $\sigma$ =0.4)    | 99.77  | 0.000 | ✓     | 100.0  | 0.000 | ✓     | 99.92  | 0.000 | ✓            | 99.90 ± 0.10                      | 0.000 ± 0.000                     |
|                                    | Noise Gaussian ( $\sigma$ =0.5)    | 99.77  | 0.000 | ✓     | 100.0  | 0.000 | ✓     | 99.92  | 0.000 | ✓            | 99.90 ± 0.10                      | 0.000 ± 0.000                     |
|                                    | Salt and Pepper                    | 95.40  | 0.015 | ✓     | 94.94  | 0.016 | ✓     | 96.03  | 0.013 | ✓            | 95.46 ± 0.45                      | 0.015 ± 0.002                     |
|                                    | Crop Height(top, 50%)              | 96.41  | 0.018 | ✓     | 94.67  | 0.021 | ✓     | 94.50  | 0.021 | ✓            | 95.19 ± 0.87                      | 0.020 ± 0.002                     |
|                                    | Crop Width (left, 25%)             | 99.61  | 0.001 | ✓     | 99.94  | 0.000 | ✓     | 99.88  | 0.000 | ✓            | 99.81 ± 0.14                      | 0.000 ± 0.000                     |
| HEVC                               | Gamma Correction ( $\gamma$ =1.25) | 99.77  | 0.000 | ✓     | 100.0  | 0.000 | ✓     | 99.92  | 0.000 | ✓            | 99.90 ± 0.10                      | 0.000 ± 0.000                     |
|                                    | Gamma Correction ( $\gamma$ =2.5)  | 94.43  | 0.015 | ✓     | 94.42  | 0.015 | ✓     | 94.41  | 0.014 | ✓            | 94.42 ± 0.01                      | 0.015 ± 0.001                     |
|                                    | Histogram Equalization             | 99.74  | 0.000 | ✓     | 99.86  | 0.000 | ✓     | 99.80  | 0.000 | ✓            | 99.80 ± 0.05                      | 0.000 ± 0.000                     |
|                                    | No Attack                          | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗            | 100.0 ± 0.00                      | 0.000 ± 0.000                     |
|                                    | Blur Average (3 × 3)               | 99.66  | 0.001 | ✓     | 98.58  | 0.001 | ✓     | 100.0  | 0.000 | ✓            | 99.41 ± 0.74                      | 0.001 ± 0.001                     |
|                                    | Blur Average (7 × 7)               | 94.58  | 0.031 | ✓     | 93.15  | 0.029 | ✓     | 93.49  | 0.035 | ✓            | 93.74 ± 0.60                      | 0.032 ± 0.003                     |
|                                    | Blur Gaussian (5 × 5)              | 99.69  | 0.002 | ✓     | 98.81  | 0.000 | ✓     | 99.90  | 0.001 | ✓            | 99.47 ± 0.46                      | 0.001 ± 0.001                     |
|                                    | Blur Gaussian (9 × 9)              | 92.76  | 0.039 | ✓     | 90.42  | 0.046 | ✓     | 94.00  | 0.036 | ✓            | 92.39 ± 1.48                      | 0.040 ± 0.004                     |
|                                    | Blur Median (3 × 3)                | 94.40  | 0.011 | ✓     | 95.38  | 0.008 | ✓     | 96.18  | 0.008 | ✓            | 95.32 ± 0.73                      | 0.009 ± 0.002                     |
|                                    | Blur Median (9 × 9)                | 100.0  | 0.000 | ✓     | 98.65  | 0.001 | ✓     | 100.0  | 0.000 | ✓            | 99.55 ± 0.64                      | 0.000 ± 0.000                     |
|                                    | Compression (Q=50)                 | 99.71  | 0.001 | ✓     | 98.67  | 0.001 | ✓     | 100.0  | 0.000 | ✓            | 99.46 ± 0.67                      | 0.001 ± 0.001                     |
|                                    | Compression (Q=70)                 | 94.56  | 0.010 | ✓     | 94.27  | 0.011 | ✓     | 95.62  | 0.010 | ✓            | 94.82 ± 0.58                      | 0.010 ± 0.001                     |
|                                    | Noise Gaussian ( $\sigma$ =0.4)    | 100.0  | 0.000 | ✓     | 98.83  | 0.000 | ✓     | 100.0  | 0.000 | ✓            | 99.61 ± 0.55                      | 0.000 ± 0.000                     |
|                                    | Noise Gaussian ( $\sigma$ =0.5)    | 100.0  | 0.000 | ✓     | 98.81  | 0.000 | ✓     | 100.0  | 0.000 | ✓            | 99.60 ± 0.56                      | 0.000 ± 0.000                     |
| AV1                                | Salt and Pepper                    | 96.49  | 0.012 | ✓     | 94.66  | 0.014 | ✓     | 95.89  | 0.013 | ✓            | 95.68 ± 0.75                      | 0.013 ± 0.001                     |
|                                    | Crop Height(top, 50%)              | 96.56  | 0.018 | ✓     | 95.19  | 0.020 | ✓     | 96.19  | 0.020 | ✓            | 95.98 ± 0.57                      | 0.019 ± 0.001                     |
|                                    | Crop Width (left, 25%)             | 99.95  | 0.000 | ✓     | 98.73  | 0.001 | ✓     | 99.94  | 0.000 | ✓            | 99.54 ± 0.57                      | 0.000 ± 0.001                     |
|                                    | Gamma Correction ( $\gamma$ =1.25) | 100.0  | 0.000 | ✓     | 98.81  | 0.000 | ✓     | 100.0  | 0.000 | ✓            | 99.60 ± 0.56                      | 0.000 ± 0.000                     |
|                                    | Gamma Correction ( $\gamma$ =2.5)  | 94.79  | 0.018 | ✓     | 93.02  | 0.016 | ✓     | 95.34  | 0.013 | ✓            | 94.38 ± 0.97                      | 0.016 ± 0.002                     |
|                                    | Histogram Equalization             | 100.0  | 0.000 | ✓     | 98.83  | 0.000 | ✓     | 99.87  | 0.000 | ✓            | 99.57 ± 0.49                      | 0.000 ± 0.000                     |
|                                    | No Attack                          | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗            | 100.0 ± 0.00                      | 0.000 ± 0.000                     |
|                                    | Blur Average (3 × 3)               | 99.58  | 0.001 | ✓     | 99.83  | 0.000 | ✓     | 99.89  | 0.000 | ✓            | 99.77 ± 0.13                      | 0.001 ± 0.000                     |
|                                    | Blur Average (7 × 7)               | 94.30  | 0.031 | ✓     | 93.71  | 0.039 | ✓     | 92.75  | 0.038 | ✓            | 93.59 ± 0.64                      | 0.036 ± 0.004                     |
|                                    | Blur Gaussian (5 × 5)              | 99.64  | 0.002 | ✓     | 99.77  | 0.002 | ✓     | 99.80  | 0.001 | ✓            | 99.74 ± 0.08                      | 0.002 ± 0.001                     |
|                                    | Blur Gaussian (9 × 9)              | 91.41  | 0.047 | ✓     | 95.19  | 0.032 | ✓     | 91.25  | 0.041 | ✓            | 92.62 ± 1.79                      | 0.040 ± 0.006                     |
|                                    | Blur Median (3 × 3)                | 93.65  | 0.011 | ✓     | 95.25  | 0.010 | ✓     | 94.20  | 0.010 | ✓            | 94.37 ± 0.65                      | 0.010 ± 0.000                     |
|                                    | Blur Median (9 × 9)                | 99.77  | 0.000 | ✓     | 99.96  | 0.000 | ✓     | 99.81  | 0.000 | ✓            | 99.85 ± 0.08                      | 0.000 ± 0.000                     |
|                                    | Compression (Q=50)                 | 100.0  | 0.000 | ✓     | 99.96  | 0.000 | ✓     | 99.96  | 0.000 | ✓            | 99.97 ± 0.02                      | 0.000 ± 0.000                     |
| Compression (Q=70)                 | 94.97                              | 0.009  | ✓     | 95.42 | 0.011  | ✓     | 95.58 | 0.008  | ✓     | 95.32 ± 0.26 | 0.009 ± 0.001                     |                                   |
| Noise Gaussian ( $\sigma$ =0.4)    | 100.0                              | 0.000  | ✓     | 99.96 | 0.000  | ✓     | 99.96 | 0.000  | ✓     | 99.97 ± 0.02 | 0.000 ± 0.000                     |                                   |
| Noise Gaussian ( $\sigma$ =0.5)    | 100.0                              | 0.000  | ✓     | 99.96 | 0.000  | ✓     | 99.96 | 0.000  | ✓     | 99.97 ± 0.02 | 0.000 ± 0.000                     |                                   |
| Salt and Pepper                    | 95.50                              | 0.015  | ✓     | 95.89 | 0.014  | ✓     | 96.03 | 0.013  | ✓     | 95.81 ± 0.22 | 0.014 ± 0.001                     |                                   |
| Crop Height(top, 50%)              | 95.08                              | 0.022  | ✓     | 96.08 | 0.020  | ✓     | 95.54 | 0.017  | ✓     | 95.57 ± 0.41 | 0.020 ± 0.002                     |                                   |
| Crop Width (left, 25%)             | 99.79                              | 0.001  | ✓     | 99.92 | 0.000  | ✓     | 99.92 | 0.000  | ✓     | 99.88 ± 0.06 | 0.000 ± 0.000                     |                                   |
| Gamma Correction ( $\gamma$ =1.25) | 100.0                              | 0.000  | ✓     | 99.96 | 0.000  | ✓     | 99.96 | 0.000  | ✓     | 99.97 ± 0.02 | 0.000 ± 0.000                     |                                   |
| Gamma Correction ( $\gamma$ =2.5)  | 93.12                              | 0.017  | ✓     | 93.71 | 0.018  | ✓     | 94.68 | 0.014  | ✓     | 93.84 ± 0.64 | 0.016 ± 0.002                     |                                   |
| Histogram Equalization             | 100.0                              | 0.000  | ✓     | 99.84 | 0.000  | ✓     | 99.87 | 0.000  | ✓     | 99.90 ± 0.07 | 0.000 ± 0.000                     |                                   |

Table 6.27: Verification and robustness results for the code-based hash-chain video watermarking model on 4K videos of H264, HEVC, and AV1 codecs across 24/30/60 FPS, depicting the mean  $\pm$  standard deviation Extraction Rate( $R_e$ ) and BER for each row.

|                                    |                                    | 24 FPS |       |       | 30 FPS |       |       | 60 FPS |       |              | Average                           |                                   |
|------------------------------------|------------------------------------|--------|-------|-------|--------|-------|-------|--------|-------|--------------|-----------------------------------|-----------------------------------|
| Attack                             |                                    | $R_e$  | BER   | T     | $R_e$  | BER   | T     | $R_e$  | BER   | T            | $\overline{R_e} \pm \sigma_{R_e}$ | $\overline{BER} \pm \sigma_{BER}$ |
| H264                               | No Attack                          | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗            | 100.0 ± 0.00                      | 0.000 ± 0.000                     |
|                                    | Blur Average (3 × 3)               | 90.96  | 0.013 | ✓     | 91.19  | 0.012 | ✓     | 99.79  | 0.001 | ✓            | 93.98 ± 5.03                      | 0.009 ± 0.007                     |
|                                    | Blur Average (7 × 7)               | 84.90  | 0.039 | ✓     | 85.23  | 0.039 | ✓     | 93.06  | 0.033 | ✓            | 87.73 ± 4.62                      | 0.037 ± 0.003                     |
|                                    | Blur Gaussian (5 × 5)              | 90.99  | 0.014 | ✓     | 88.51  | 0.032 | ✓     | 99.82  | 0.001 | ✓            | 93.11 ± 5.64                      | 0.016 ± 0.016                     |
|                                    | Blur Gaussian (9 × 9)              | 87.71  | 0.029 | ✓     | 87.69  | 0.027 | ✓     | 91.58  | 0.049 | ✓            | 88.99 ± 2.24                      | 0.035 ± 0.012                     |
|                                    | Blur Median (3 × 3)                | 87.73  | 0.023 | ✓     | 86.42  | 0.017 | ✓     | 96.09  | 0.015 | ✓            | 90.08 ± 4.99                      | 0.018 ± 0.004                     |
|                                    | Blur Median (9 × 9)                | 90.99  | 0.013 | ✓     | 91.08  | 0.013 | ✓     | 99.91  | 0.001 | ✓            | 93.99 ± 5.06                      | 0.009 ± 0.006                     |
|                                    | Compression (Q=50)                 | 91.04  | 0.014 | ✓     | 91.12  | 0.013 | ✓     | 99.66  | 0.001 | ✓            | 93.94 ± 4.97                      | 0.009 ± 0.006                     |
|                                    | Compression (Q=70)                 | 85.44  | 0.040 | ✓     | 85.62  | 0.039 | ✓     | 92.40  | 0.021 | ✓            | 87.82 ± 3.99                      | 0.033 ± 0.010                     |
|                                    | Noise Gaussian ( $\sigma=0.4$ )    | 91.28  | 0.013 | ✓     | 91.27  | 0.012 | ✓     | 99.91  | 0.001 | ✓            | 94.15 ± 4.95                      | 0.009 ± 0.006                     |
|                                    | Noise Gaussian ( $\sigma=0.5$ )    | 91.20  | 0.012 | ✓     | 91.31  | 0.012 | ✓     | 99.91  | 0.001 | ✓            | 94.14 ± 4.96                      | 0.009 ± 0.006                     |
|                                    | Salt and Pepper                    | 87.19  | 0.030 | ✓     | 87.34  | 0.030 | ✓     | 96.54  | 0.014 | ✓            | 90.36 ± 4.87                      | 0.025 ± 0.009                     |
|                                    | Crop Height (top, 50%)             | 87.68  | 0.032 | ✓     | 87.90  | 0.031 | ✓     | 95.92  | 0.018 | ✓            | 90.50 ± 4.41                      | 0.027 ± 0.007                     |
|                                    | Crop Width (left, 25%)             | 91.09  | 0.013 | ✓     | 91.23  | 0.012 | ✓     | 99.91  | 0.001 | ✓            | 94.08 ± 4.97                      | 0.009 ± 0.006                     |
| HEVC                               | Gamma Correction ( $\gamma=1.25$ ) | 90.99  | 0.013 | ✓     | 91.17  | 0.013 | ✓     | 99.88  | 0.001 | ✓            | 94.01 ± 4.99                      | 0.009 ± 0.006                     |
|                                    | Gamma Correction ( $\gamma=2.5$ )  | 84.74  | 0.040 | ✓     | 85.19  | 0.041 | ✓     | 98.45  | 0.003 | ✓            | 89.46 ± 7.63                      | 0.028 ± 0.019                     |
|                                    | Histogram Equalization             | 90.77  | 0.015 | ✓     | 90.49  | 0.016 | ✓     | 99.18  | 0.004 | ✓            | 93.48 ± 4.96                      | 0.012 ± 0.006                     |
|                                    | No Attack                          | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗            | 100.0 ± 0.00                      | 0.000 ± 0.000                     |
|                                    | Blur Average (3 × 3)               | 89.74  | 0.016 | ✓     | 88.79  | 0.016 | ✓     | 99.75  | 0.001 | ✓            | 92.76 ± 6.07                      | 0.011 ± 0.009                     |
|                                    | Blur Average (7 × 7)               | 83.93  | 0.041 | ✓     | 82.96  | 0.043 | ✓     | 93.12  | 0.033 | ✓            | 86.67 ± 5.61                      | 0.039 ± 0.005                     |
|                                    | Blur Gaussian (5 × 5)              | 89.71  | 0.018 | ✓     | 88.65  | 0.017 | ✓     | 99.79  | 0.002 | ✓            | 92.72 ± 6.13                      | 0.012 ± 0.009                     |
|                                    | Blur Gaussian (9 × 9)              | 85.81  | 0.032 | ✓     | 85.15  | 0.032 | ✓     | 92.38  | 0.045 | ✓            | 87.78 ± 4.05                      | 0.036 ± 0.007                     |
|                                    | Blur Median (3 × 3)                | 87.68  | 0.023 | ✓     | 86.31  | 0.023 | ✓     | 96.18  | 0.014 | ✓            | 90.06 ± 5.02                      | 0.020 ± 0.005                     |
|                                    | Blur Median (9 × 9)                | 89.61  | 0.017 | ✓     | 88.67  | 0.017 | ✓     | 99.88  | 0.001 | ✓            | 92.72 ± 6.13                      | 0.012 ± 0.009                     |
|                                    | Compression (Q=50)                 | 89.69  | 0.017 | ✓     | 88.69  | 0.017 | ✓     | 99.75  | 0.001 | ✓            | 92.71 ± 6.10                      | 0.012 ± 0.009                     |
|                                    | Compression (Q=70)                 | 84.06  | 0.043 | ✓     | 83.35  | 0.043 | ✓     | 92.11  | 0.022 | ✓            | 86.51 ± 4.75                      | 0.036 ± 0.012                     |
|                                    | Noise Gaussian ( $\sigma=0.4$ )    | 89.82  | 0.016 | ✓     | 88.77  | 0.016 | ✓     | 99.88  | 0.001 | ✓            | 92.82 ± 6.12                      | 0.011 ± 0.009                     |
|                                    | Noise Gaussian ( $\sigma=0.5$ )    | 89.90  | 0.016 | ✓     | 88.85  | 0.016 | ✓     | 99.88  | 0.001 | ✓            | 92.88 ± 6.10                      | 0.011 ± 0.009                     |
| AV1                                | Salt and Pepper                    | 86.02  | 0.034 | ✓     | 84.96  | 0.034 | ✓     | 95.31  | 0.019 | ✓            | 88.76 ± 5.73                      | 0.029 ± 0.008                     |
|                                    | Crop Height (top, 50%)             | 85.94  | 0.035 | ✓     | 85.52  | 0.035 | ✓     | 95.43  | 0.021 | ✓            | 88.96 ± 5.54                      | 0.030 ± 0.008                     |
|                                    | Crop Width (left, 25%)             | 89.79  | 0.017 | ✓     | 88.79  | 0.016 | ✓     | 99.83  | 0.001 | ✓            | 92.80 ± 6.14                      | 0.011 ± 0.009                     |
|                                    | Gamma Correction ( $\gamma=1.25$ ) | 89.66  | 0.017 | ✓     | 88.71  | 0.017 | ✓     | 99.85  | 0.001 | ✓            | 92.74 ± 6.14                      | 0.012 ± 0.009                     |
|                                    | Gamma Correction ( $\gamma=2.5$ )  | 83.96  | 0.044 | ✓     | 82.88  | 0.044 | ✓     | 98.12  | 0.004 | ✓            | 88.32 ± 8.48                      | 0.031 ± 0.023                     |
|                                    | Histogram Equalization             | 88.96  | 0.021 | ✓     | 88.28  | 0.020 | ✓     | 99.03  | 0.005 | ✓            | 92.09 ± 5.96                      | 0.015 ± 0.009                     |
|                                    | No Attack                          | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗     | 100.0  | 0.000 | ✗            | 100.0 ± 0.00                      | 0.000 ± 0.000                     |
|                                    | Blur Average (3 × 3)               | 88.15  | 0.018 | ✓     | 90.17  | 0.016 | ✓     | 99.53  | 0.001 | ✓            | 92.62 ± 6.07                      | 0.012 ± 0.009                     |
|                                    | Blur Average (7 × 7)               | 83.44  | 0.044 | ✓     | 84.42  | 0.042 | ✓     | 93.07  | 0.034 | ✓            | 86.98 ± 5.30                      | 0.040 ± 0.006                     |
|                                    | Blur Gaussian (5 × 5)              | 88.12  | 0.020 | ✓     | 90.12  | 0.017 | ✓     | 99.49  | 0.002 | ✓            | 92.58 ± 6.10                      | 0.013 ± 0.010                     |
|                                    | Blur Gaussian (9 × 9)              | 84.87  | 0.033 | ✓     | 86.17  | 0.032 | ✓     | 93.30  | 0.039 | ✓            | 88.11 ± 4.50                      | 0.035 ± 0.004                     |
|                                    | Blur Median (3 × 3)                | 84.97  | 0.029 | ✓     | 85.58  | 0.023 | ✓     | 95.75  | 0.015 | ✓            | 88.77 ± 5.71                      | 0.022 ± 0.006                     |
|                                    | Blur Median (9 × 9)                | 88.07  | 0.019 | ✓     | 90.12  | 0.017 | ✓     | 99.62  | 0.001 | ✓            | 92.60 ± 6.14                      | 0.012 ± 0.009                     |
|                                    | Compression (Q=50)                 | 87.99  | 0.020 | ✓     | 90.17  | 0.017 | ✓     | 99.72  | 0.001 | ✓            | 92.63 ± 6.15                      | 0.013 ± 0.010                     |
| Compression (Q=70)                 | 82.42                              | 0.045  | ✓     | 84.46 | 0.044  | ✓     | 93.12 | 0.018  | ✓     | 86.67 ± 5.54 | 0.036 ± 0.014                     |                                   |
| Noise Gaussian ( $\sigma=0.4$ )    | 88.31                              | 0.018  | ✓     | 90.44 | 0.016  | ✓     | 99.72 | 0.001  | ✓     | 92.82 ± 6.12 | 0.012 ± 0.009                     |                                   |
| Noise Gaussian ( $\sigma=0.5$ )    | 88.26                              | 0.018  | ✓     | 90.31 | 0.016  | ✓     | 99.72 | 0.001  | ✓     | 92.76 ± 6.13 | 0.012 ± 0.009                     |                                   |
| Salt and Pepper                    | 84.59                              | 0.036  | ✓     | 86.44 | 0.034  | ✓     | 95.66 | 0.017  | ✓     | 88.90 ± 5.67 | 0.029 ± 0.008                     |                                   |
| Crop Height (top, 50%)             | 84.87                              | 0.037  | ✓     | 85.69 | 0.035  | ✓     | 96.07 | 0.017  | ✓     | 88.88 ± 5.87 | 0.030 ± 0.010                     |                                   |
| Crop Width (left, 25%)             | 88.20                              | 0.019  | ✓     | 90.23 | 0.016  | ✓     | 99.68 | 0.001  | ✓     | 92.70 ± 6.11 | 0.012 ± 0.009                     |                                   |
| Gamma Correction ( $\gamma=1.25$ ) | 88.12                              | 0.019  | ✓     | 90.10 | 0.017  | ✓     | 99.72 | 0.001  | ✓     | 92.65 ± 6.13 | 0.012 ± 0.009                     |                                   |
| Gamma Correction ( $\gamma=2.5$ )  | 82.19                              | 0.045  | ✓     | 83.79 | 0.043  | ✓     | 97.64 | 0.004  | ✓     | 87.87 ± 7.96 | 0.031 ± 0.023                     |                                   |
| Histogram Equalization             | 88.13                              | 0.020  | ✓     | 89.10 | 0.021  | ✓     | 98.83 | 0.005  | ✓     | 92.02 ± 5.91 | 0.015 ± 0.009                     |                                   |

and F1-score values remaining approximately equal to 100%.

Table 6.26 presents the robustness results obtained for 2K videos encoded using H.264, HEVC, and AV1 at 24, 30, and 60 FPS. As we can observe, the proposed model demonstrates high robustness across all codecs and frame rates, with extraction rates remaining close to 100% and BER values remaining very low in the majority of attack scenarios. Furthermore, the small standard deviations indicate limited sensitivity to frame-rate variations. The highest robustness is observed under Gaussian noise, light filtering operations, histogram equalization, moderate gamma correction ( $\gamma = 1.25$ ), compression with  $Q = 50$ , and horizontal cropping, where extraction rates frequently reach 100% and BER values approach zero. The most challenging attacks correspond to large-kernel filtering, strong gamma correction ( $\gamma = 2.5$ ), and aggressive compression ( $Q = 70$ ), which reduce extraction rates to approximately 90% – 93%. Nevertheless, successful watermark extraction is achieved in all cases. The three codecs exhibit similar robustness characteristics, with HEVC and AV1 demonstrating slightly improved performance under severe distortions. Finally, all attack scenarios achieved a 100% tamper detection rate, while Accuracy, Precision, Recall, and F1-score values remained approximately equal to 100%, confirming the effectiveness of the proposed authenticity verification mechanism for high-resolution video content.

Table 6.26 presents the robustness results obtained for 4K videos encoded using H.264, HEVC, and AV1 at 24, 30, and 60 FPS. Despite the significantly increased resolution, the proposed model maintains high robustness across all evaluated codecs and attack scenarios, with extraction rates generally remaining above 85% and BER values remaining relatively low.

The highest robustness is observed under Gaussian noise, light filtering operations, histogram equalization, moderate gamma correction ( $\gamma = 1.25$ ), and small cropping attacks, where extraction rates frequently exceed 92% while BER values remain close to zero. The most challenging attack scenarios correspond to large-kernel Gaussian blur ( $9 \times 9$ ), aggressive compression ( $Q = 70$ ), and strong gamma correction ( $\gamma = 2.5$ ), which reduce extraction rates to approximately 82% – 88%. Nevertheless, successful watermark extraction is achieved in all cases. Furthermore, the small standard deviations across the evaluated frame rates indicate limited sensitivity to temporal variations, confirming stable performance at 24, 30, and 60 FPS. The three codecs exhibit similar robustness characteristics, with AV1 producing slightly lower extraction rates under severe distortions while remaining within a reliable operating range. Finally, all attack scenarios achieved a 100% tamper detection rate, with Accuracy,

Table 6.28: Robustness and verification evaluation of the code-based hash-chain video watermarking model under spatial, temporal, compression, and combined transformations occurring after video sharing or transmission, i.e., Frame Rate Reduction (FRR), Frame Rate Increase (FRI), Spatial Downscale (SD), Spatial Upscale (SU), Re-Encoding (RE), Frame Dropping (FD), and Frame Swap (FS).

| Resolution | Transformation Type           | $\overline{R_e}$ (%) | $\overline{BER}$ | Tampering Detected |
|------------|-------------------------------|----------------------|------------------|--------------------|
| 1080p      | FRR (24 fps)                  | 99.86                | 0.00081          | ✓                  |
| 1080p      | FRI (60 fps)                  | 99.87                | 0.00083          | ✓                  |
| 720p       | SD (decrease $2.25\times$ )   | 14.91                | 0.40             | ✓                  |
| 2K         | SU (increase $1.78\times$ )   | 12.29                | 0.46             | ✓                  |
| 1080p      | RE HEVC                       | 96.35                | 0.0053           | ✓                  |
| 1080p      | RE AVI                        | 98.41                | 0.0045           | ✓                  |
| 1080p      | RE MPEG-4 (1.5Mbps)           | 62.12                | 0.13             | ✓                  |
| 1080p      | RE MPEG-4 (5.0Mbps)           | 95.80                | 0.006            | ✓                  |
| 1080p      | RE MPEG-4 (6.0Mbps)           | 97.70                | 0.004            | ✓                  |
| 1080p      | RE MPEG-4 (8.0Mbps)           | 98.50                | 0.002            | ✓                  |
| 1080p      | FD (20%)                      | 99.92                | 0.0002           | ✓                  |
| 1080p      | FD (30%)                      | 99.91                | 0.0003           | ✓                  |
| 1080p      | FD (50%)                      | 99.91                | 0.0003           | ✓                  |
| 1080p      | FS (20%)                      | 99.95                | 0.0002           | ✓                  |
| 1080p      | FS (30%)                      | 99.95                | 0.0002           | ✓                  |
| 1080p      | FS (50%)                      | 99.95                | 0.0002           | ✓                  |
| 720p       | SD 720p + FRR 24fps + RE HEVC | 26.03                | 0.44             | ✓                  |
| 2K         | SU 1440p + FRI 60fps + RE AVI | 12.85                | 0.43             | ✓                  |

Precision, Recall, and F1-score values remaining approximately equal to 100%.

The results presented in Table 6.28 demonstrate the robustness of the proposed model under spatial, temporal, compression, and combined transformations commonly encountered during video sharing and transmission. Temporal modifications,

including FRR, FRI, FD, and FS, have minimal impact on watermark extraction, achieving extraction rates above 99.8% with BER values close to zero. Similarly, re-encoding operations using HEVC, AVI, and moderate MPEG-4 compression preserve high extraction performance, with extraction rates generally exceeding 95%. In contrast, spatial scaling operations, including downscaling and upscaling, constitute the most challenging attack category, reducing extraction rates to approximately 12% – 15% due to the alteration of the underlying spatial structure and frequency-domain representation of the video frames. Combined transformations that include spatial scaling exhibit similar degradation effects. Nevertheless, despite the reduction in extraction performance, all attack scenarios were successfully identified by the proposed authenticity verification mechanism.

Overall, the results demonstrate that the proposed framework exhibits strong robustness against temporal and compression-based transformations while remaining capable of reliably detecting severe geometric modifications. The achieved results validate the effectiveness of the proposed cross-modal verification strategy for authenticity and integrity preservation in digital video content.

#### **6.7.4 Video: Integrity Attestation**

##### **Cross-Frame Permutation Based Video Watermarking Model**

As illustrated in Figure 6.11, integrity-blocks identified as tampered are highlighted in red, enabling the visual localization of manipulated regions within the video frame. When a tampering operation affects multiple integrity-blocks, all corresponding blocks are marked, providing a clear representation of the extent of the modification.

Table 6.29 summarizes the average tamper detection and localization performance of the proposed framework under the single-frame tampering scenario. The obtained results demonstrate near-perfect performance across all evaluated metrics, confirming the effectiveness of the proposed integrity-preservation mechanism. For Addition and Copy-Move attacks, the proposed model achieves perfect performance, with Precision, Recall, F1-score, and IoU values equal to 1.0. These results indicate that all tampered regions were correctly detected and accurately localized without generating false-positive or false-negative detections.

Similarly, Removal attacks exhibit excellent performance, with Recall remaining equal to 1.0 and Precision, F1-score, and IoU values remaining above 0.97. The

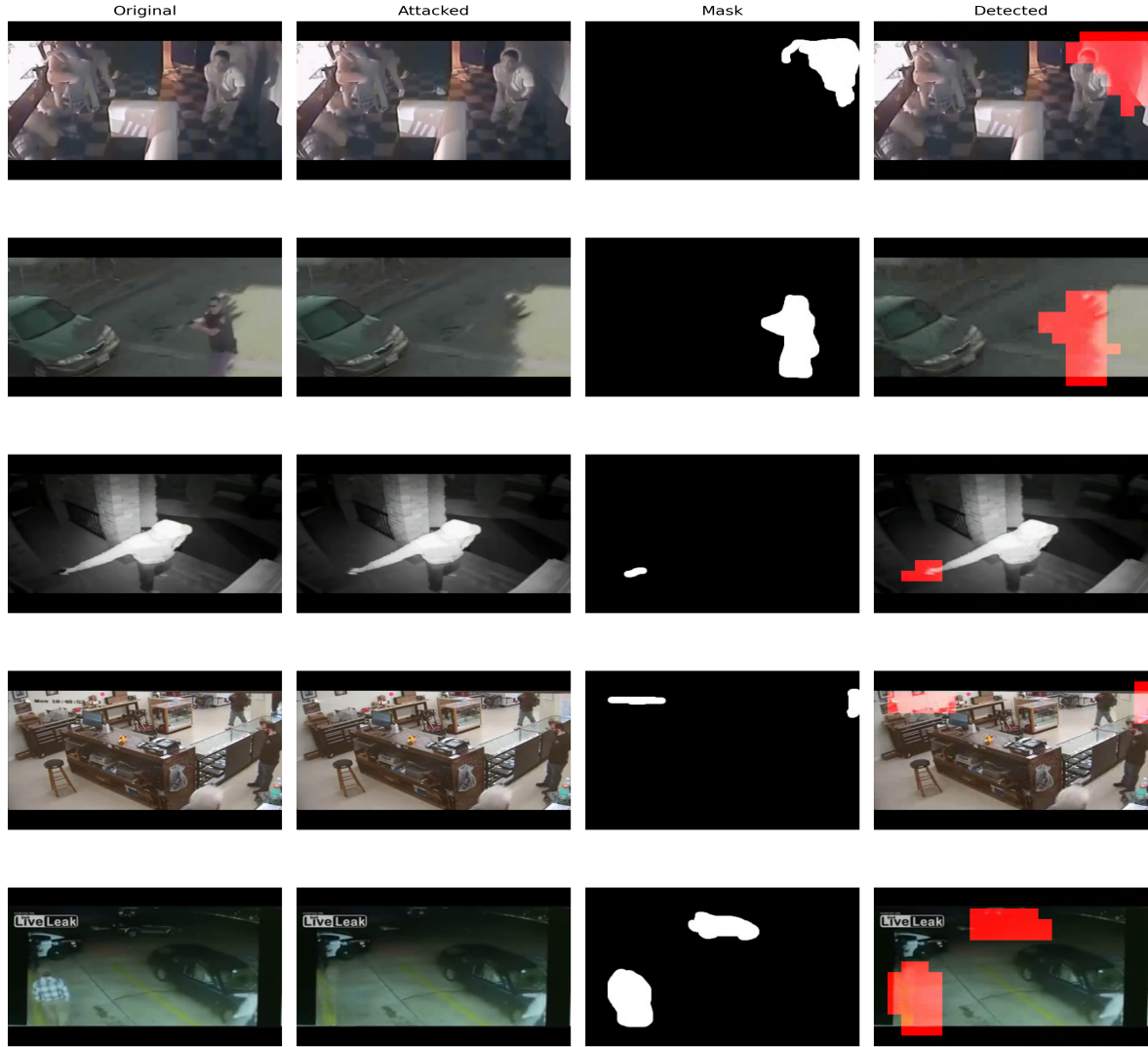


Figure 6.11: Example of Tampered Frames and Tamper Detection Results per frame under removal attack.

slight reduction in Precision can be attributed to a limited number of false-positive detections occurring at the boundaries of the manipulated regions. Nevertheless, the high IoU values confirm the accurate localization of the tampered areas.

Table 6.30 presents the average tamper detection and localization performance of the proposed model under multi-frame Addition, Removal, and Copy-Move attacks. The results demonstrate consistently high Accuracy and Precision values exceeding 0.99 across all attack categories. The highest performance is achieved for Removal attacks, while Copy-Move attacks exhibit slightly lower Recall and IoU values due to the increased difficulty of accurately localizing duplicated regions. Nevertheless, the



Table 6.29: Average performance of the proposed cross-frame permutation based video watermarking model under single-frame tampering for different attack types. Metrics are averaged across all evaluated videos.

| Attack       | Accuracy | Precision | Recall | F1 Score | IoU   |
|--------------|----------|-----------|--------|----------|-------|
| SF Removal   | 0.994    | 0.970     | 1.000  | 0.984    | 0.974 |
| SF Addition  | 1.000    | 1.000     | 1.000  | 1.000    | 1.000 |
| SF Copy-Move | 1.000    | 1.000     | 1.000  | 1.000    | 1.000 |

Table 6.30: Average tamper detection and localization performance under multi-frame addition, removal and copy-move attacks for the cross-frame permutation based video watermarking model.

| Attack       | Accuracy $\pm \sigma$           | Precision $\pm \sigma$ | Recall $\pm \sigma$ | F1 – score $\pm \sigma$ | IoU $\pm \sigma$    |
|--------------|---------------------------------|------------------------|---------------------|-------------------------|---------------------|
| MF Addition  | $0.9900 \pm 1.2 \times 10^{-5}$ | $0.9900 \pm 0.0010$    | $0.9800 \pm 0.0040$ | $0.9900 \pm 0.0020$     | $0.9800 \pm 0.0040$ |
| MF Removal   | $0.9999 \pm 2.0 \times 10^{-5}$ | $0.9980 \pm 0.0020$    | $0.9810 \pm 0.0100$ | $0.9910 \pm 0.0050$     | $0.9810 \pm 0.0080$ |
| MF Copy-Move | $0.9998 \pm 1.5 \times 10^{-5}$ | $0.9980 \pm 0.0020$    | $0.9590 \pm 0.0500$ | $0.9890 \pm 0.0060$     | $0.9580 \pm 0.0500$ |

obtained F1-scores remain close to 0.99, confirming the effectiveness of the proposed method in detecting and localizing different types of video tampering.

Overall, the obtained results demonstrate that the proposed framework is capable of reliably detecting and localizing a wide range of tampering operations. While Accuracy remains close to 1.0 for all attack categories, metrics such as Precision, Recall, F1-score, and IoU provide a more representative assessment of performance, as they directly evaluate the quality of tamper localization in the presence of class imbalance between tampered and untampered regions.

### 6.7.5 Video: Observations

For the code-based hash-chain model, the obtained results indicate that the proposed watermark representation remains highly robust against a wide range of signal-processing, compression, geometric, and temporal attacks. Across all evaluated resolutions, codecs, and frame rates, extraction rates remained consistently high while BER values remained low, demonstrating the stability of the proposed embedding strategy.

Furthermore, the obtained results reveal that the model is largely independent of the selected codec and frame rate, with resolution being the dominant factor affecting computational complexity. The utilization of a global hash-chain representation embedded into the audio stream further strengthens the security of the framework by establishing a dependency between the visual and auditory components of the video object. Consequently, modifications affecting

Regarding the cross-frame permutation-based model, the experimental results demonstrate that the proposed model achieves high perceptual quality despite the relatively large integrity payload. More specifically, PSNR values ranged from 41.23 dB to 50.57 dB, with an average value of 46.11 dB, while the average SSIM reached 0.987. These measurements indicate that the embedded integrity information introduces only limited visual distortion and preserves the visual characteristics of the protected video content.

An important observation concerns the effectiveness of the introduced cross-frame verification relationships. Unlike conventional block-based integrity-preservation approaches, the proposed framework establishes dependencies among integrity-blocks located in different frames. As a result, manipulations affecting individual frames, groups of frames, or temporally distant video regions propagate through the verification process and can be reliably detected. The experimental evaluation confirmed that the framework successfully detects object insertion, object deletion, copy-move manipulations.

The run-time evaluation additionally revealed favorable scalability characteristics for both frameworks. For the authenticity-preservation model, the processing time increased predictably with video resolution while remaining largely unaffected by codec selection and frame rate variations. Furthermore, the parallel implementation achieved substantial reductions in execution time, demonstrating the suitability of the proposed framework for deployment on multi-core architectures. For the integrity-preservation model, a strong positive linear relationship was observed between permutation size and embedding time, confirming the predictability and scalability of the proposed cross-frame verification architecture.

Overall, the obtained results demonstrate that the proposed video watermarking models successfully combine imperceptibility, robustness, authenticity verification, integrity preservation, and tamper localization capabilities. The complementary nature of the two approaches enables the protection of digital videos against both content-

preservation attacks and malicious manipulations, making them suitable for multimedia forensics, surveillance systems, evidential video management, and authenticity-preservation applications.

## 6.8 Experimental Results on Documents

Next we present the experimental results regarding the proposed PDF authenticity and integrity preservation model. Specifically, the evaluation focuses on the computational efficiency, visual quality, authenticity verification, and tamper detection performance of the proposed watermarking models under different attack categories.

The following PDF model is experimentally evaluated:

- **PDF Watermarking Model utilizing QR-Code Text-Color Embedding**

### **PDF Watermarking Model utilizing QR-Code Text-Color Embedding - Methodology**

For the experimental evaluation of the proposed PDF watermarking model, a dataset consisting of both single-page and multi-page PDF documents was utilized. The PDF files were generated from text-based documents and exported to PDF format, while the corresponding QR Codes were additionally stored in PNG format to facilitate quality and robustness analysis. The evaluation was divided into two main categories: quality assessment and robustness assessment. For the quality evaluation, the extracted QR Code was compared against the originally embedded QR Code using PSNR and SSIM measurements, while its functionality was verified through successful decoding of the associated URL.

To evaluate the robustness and fragility characteristics of the proposed model, two categories of attacks were considered. The first category included benign PDF processing operations, such as markup insertion, digital signatures, page rotation, metadata deletion, cropping, margin modifications, scaling, and compression. The second category included malicious content-modification attacks, namely PDF-to-image conversion, text removal, text modification, and text insertion. For each attack scenario, the modified PDF document was provided as input to the extraction procedure and the resulting QR Code was exported in PNG format. The functionality, quality, and recoverability of the extracted QR Code were subsequently evaluated in order to assess

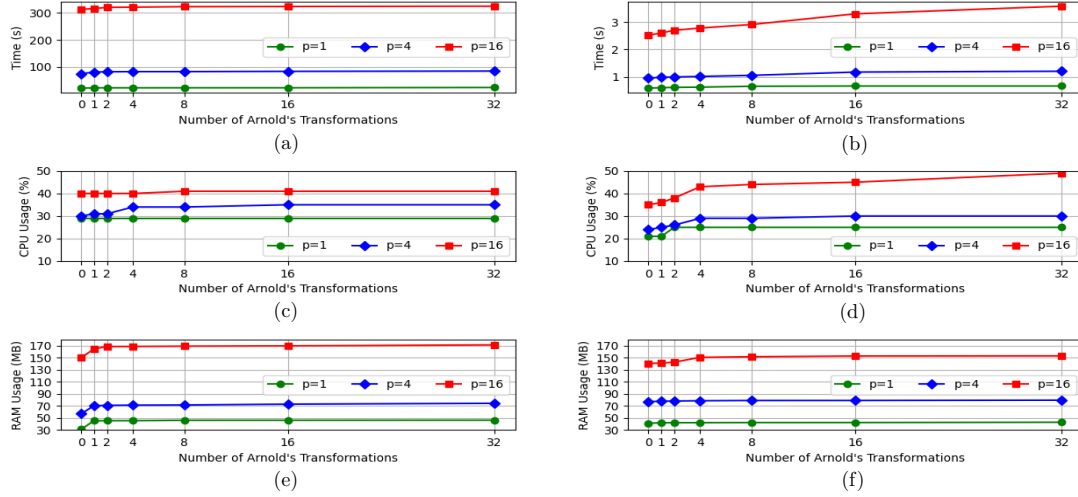


Figure 6.12: Performance measurements during the embedding and the extraction phases of the PDF watermarking model showing respectively the required time during the embedding (a) and the extraction (b) phases, the CPU utilization during the embedding (c) and the extraction (d) phases, and the RAM utilization during the embedding (e) and the extraction (f) phases.

the effectiveness of the proposed model in preserving document authenticity while enabling reliable tamper detection. The proposed repetitive watermarking technique was developed in Python 3.10 and the corresponding evaluation experiments were conducted in a commodity computer with 9<sup>th</sup> Gen. Intel i3-9500 (6 cores - 6 threads) and 8 GB of RAM running over Windows 10.

### 6.8.1 Documents: Run-time Efficiency

The performance measurements presented in Figure 6.12 demonstrate the computational efficiency and scalability of the proposed PDF watermarking model. As expected, the embedding and extraction times increase with the number of pages, since a larger number of text elements must be processed. For example, the embedding time increases from approximately 20 seconds for single-page documents to more than 300 seconds for 16-page documents, while the corresponding memory consumption increases proportionally.

The deployment of Arnold's transformation has only a limited impact on the overall computational requirements. For a fixed number of pages, increasing the number

Table 6.31: Imperceptibility metrics for all the PDF pages of the proposed PDF watermarking model.

| PAGES     | PSNR     | SSIM    | Functional QR-Code | Authenticity Attestation | Integrity Attestation |
|-----------|----------|---------|--------------------|--------------------------|-----------------------|
| ( $p_1$ ) | 57.53051 | 0.99986 | ✓                  | ✓                        | ✓                     |
| ( $p_2$ ) | 57.46688 | 0.99986 | ✓                  | ✓                        | ✓                     |
| ( $p_3$ ) | 57.59544 | 0.99986 | ✓                  | ✓                        | ✓                     |
| ( $p_4$ ) | 57.79071 | 0.99986 | ✓                  | ✓                        | ✓                     |
| AVERAGE   | 57.59588 | 0.99986 | 100%               | Authentic                | Not Tampered          |

of Arnold iterations produces only minor variations in execution time, CPU utilization, and memory consumption. This behavior is mainly attributed to the relatively small dimensions of the embedded QR Code images, which significantly reduce the computational cost of the transformation process.

Furthermore, CPU utilization remains moderate throughout the experiments, while memory consumption scales predictably with the document size. These observations indicate that the proposed model exhibits stable computational behavior and good scalability characteristics. Overall, the obtained results demonstrate that the framework can efficiently process small and medium-sized PDF documents while maintaining predictable resource requirements for larger documents.

## 6.8.2 Documents: Quality and Imperceptibility

Table 6.31 presents the imperceptibility and functionality results obtained for the proposed PDF watermarking model. As can be observed, all evaluated pages achieve PSNR values above 57 dB and SSIM values equal to 0.99986, indicating that the color modifications introduced during the QR Code embedding process are visually imperceptible and do not affect the appearance of the PDF document.

Furthermore, the extracted QR Codes remain fully functional for all evaluated pages, successfully decoding to the embedded URL. The results also demonstrate successful authenticity attestation and integrity verification for every page, confirming that the embedded watermark information can be reliably extracted and validated in the absence of tampering. Overall, the obtained measurements verify that the proposed framework preserves the visual quality of the PDF documents while simultaneously enabling reliable authenticity and integrity attestation.

Table 6.32: Quality, functionality, and authenticity-preservation results of the extracted QR Code under common PDF processing operations.

| Attack                       | PSNR     | SSIM   | BER    | NC     | MSE    | Functional QR-Code | Authenticity Preservation |
|------------------------------|----------|--------|--------|--------|--------|--------------------|---------------------------|
| No Attack Performed          | >68.8880 | 1.0000 | .0000  | 1.0000 | 0.0000 | ✓                  | ✓                         |
| Adding Markup Text           | >68.8880 | 1.0000 | 0.0000 | 1.0000 | 0.0000 | ✓                  | ✓                         |
| Digital Signature            | >68.8880 | 1.0000 | 0.0000 | 1.0000 | 0.0000 | ✓                  | ✓                         |
| 180° Page Rotation           | >68.8880 | 1.0000 | 0.0000 | 1.0000 | 0.0000 | ✓                  | ✓                         |
| Metadata Deletion            | >68.8880 | 1.0000 | 0.0000 | 1.0000 | 0.0000 | ✓                  | ✓                         |
| Horizontal Page Crop 25%     | 68.8880  | 0.9990 | 0.0000 | 0.9937 | 0.0084 | ✓                  | ✓                         |
| Horizontal Page Crop 50%     | 68.8880  | 0.9990 | 0.0000 | 0.9937 | 0.0084 | ✓                  | ✓                         |
| Horizontal Page Crop 75%     | 68.8880  | 0.9990 | 0.0000 | 0.9937 | 0.0084 | ✓                  | ✓                         |
| Vertical Page Crop 25%       | 68.8880  | 0.9990 | 0.0000 | 0.9937 | 0.0084 | ✓                  | ✓                         |
| Vertical Page Crop 50%       | 68.8880  | 0.9990 | 0.0000 | 0.9937 | 0.0084 | ✓                  | ✓                         |
| Vertical Page Crop 75%       | 68.8880  | 0.9990 | 0.0000 | 0.9937 | 0.0084 | ✓                  | ✓                         |
| Change Margins (Fit to Text) | >68.8880 | 1.0000 | 0.0000 | 1.0000 | 0.0000 | ✓                  | ✓                         |
| Change Margins (Executive)   | >68.8880 | 1.0000 | 0.0000 | 1.0000 | 0.0000 | ✓                  | ✓                         |
| Change Margins (A1)          | >68.8880 | 1.0000 | 0.0000 | 1.0000 | 0.0000 | ✓                  | ✓                         |
| Scale Down                   | 68.8880  | 0.9990 | 0.0000 | 0.9937 | 0.0084 | ✓                  | ✓                         |
| Scale Up                     | 68.8880  | 0.9990 | 0.0000 | 0.9937 | 0.0084 | ✓                  | ✓                         |
| Compression (Low)            | 68.8880  | 0.9990 | 0.0000 | 0.9937 | 0.0084 | ✓                  | ✓                         |
| Compression (Medium)         | 68.8880  | 0.9990 | 0.0000 | 0.9937 | 0.0084 | ✓                  | ✓                         |
| Compression (High)           | 68.8880  | 0.9990 | 0.0000 | 0.9937 | 0.0084 | ✓                  | ✓                         |
| Compression (72 ppi)         | >68.8880 | 1.0000 | 0.0000 | 1.0000 | 0.0000 | ✓                  | ✓                         |

### 6.8.3 Documents: Authenticity Verification

Table 6.32 presents the quality and authenticity-preservation results obtained after applying various benign PDF processing operations to the watermarked documents. Overall, the extracted QR Code remains fully functional and authenticity attestation is successfully achieved in all evaluated scenarios.

Operations such as markup insertion, digital signatures, page rotation, metadata deletion, and margin modifications produce results identical to the attack-free case, with PSNR and SSIM values remaining at their maximum levels and BER equal to zero. Similarly, page cropping, scaling, and compression introduce only negligible quality degradation, as indicated by the slightly reduced SSIM and NC values. Nevertheless, BER remains equal to zero and the extracted QR Code continues to decode successfully.

These results demonstrate that the proposed framework is highly robust against

Table 6.33: Quality, functionality, and authenticity-preservation results of the extracted QR Code after the performance of various content modification attacks.

| Attack                                | PSNR     | SSIM   | BER    | NC      | MSE    | Functional QR-Code | Tamper Detection |
|---------------------------------------|----------|--------|--------|---------|--------|--------------------|------------------|
| No Attack Performed                   | >68.8880 | 1.0000 | 0.0000 | 1.0000  | 0.0000 | ✓                  | ✗                |
| Image Conversion (No OCR)             | 49.9644  | 0.0000 | 0.6556 | -1.0000 | 0.6556 | ✗                  | ✓                |
| Remove Text from Document (Beginning) | 51.5987  | 0.0141 | 0.4500 | 0.6561  | 0.4500 | ✗                  | ✓                |
| Remove Text from Document (Middle)    | 53.0847  | 0.2575 | 0.3196 | 0.7563  | 0.3196 | ✗                  | ✓                |
| Remove Text from Document (End)       | 68.8880  | 0.442  | 0.2342 | 0.81341 | 0.2492 | ✗                  | ✓                |
| Modify Text in Document (Beginning)   | 51.6025  | 0.0141 | 0.4496 | 0.6565  | 0.4496 | ✗                  | ✓                |
| Modify Text in Document (Middle)      | 53.0902  | 0.2586 | 0.3192 | 0.7567  | 0.3192 | ✗                  | ✓                |
| Modify Text in Document (End)         | 54.1600  | 0.9205 | 0.0288 | 0.9781  | 0.0288 | ✗                  | ✓                |
| Add Text in Document (Beginning)      | 51.5987  | 0.0121 | 0.4500 | 0.6565  | 0.4500 | ✗                  | ✓                |
| Add Text in Document (Middle)         | 54.1863  | 0.4192 | 0.2480 | 0.8095  | 0.2480 | ✗                  | ✓                |
| Add Text in Document (End)            | 51.5987  | 0.0155 | 0.4500 | 0.6546  | 0.4500 | ✗                  | ✓                |

common document processing operations. Even under aggressive compression and scaling scenarios, the extracted QR Code preserves both its functionality and authenticity information. Consequently, the proposed PDF watermarking framework can reliably preserve document authenticity while remaining resilient to routine document editing, storage, and transmission procedures.

#### 6.8.4 Documents: Integrity Attestation

Table 6.33 presents the impact of various content-modification attacks on the extracted QR Code. In contrast to the benign processing operations discussed previously, all content-modification attacks result in a non-functional QR Code and successful tamper detection, demonstrating the fragility of the proposed framework with respect to document content alterations.

The PDF-to-image conversion attack without OCR exhibits the highest degradation, yielding the lowest PSNR and SSIM values together with the highest BER, indicating severe corruption of the embedded information. Similarly, text removal, modification, and addition attacks significantly affect the extracted QR Code, with the severity of the degradation depending on the position of the modification within the document.

In general, modifications occurring at the beginning or middle of the document introduce higher BER values and lower similarity measurements compared to mod-

ifications performed near the end of the document. This behavior is expected, since changes affecting a larger portion of the text used during the embedding process alter a greater number of embedded QR-Code elements. Although attacks targeting the end of the document produce less degradation and achieve higher PSNR, SSIM, and NC values, the extracted QR Code remains non-functional and tampering is still successfully detected.

Overall, the results confirm that the proposed PDF watermarking framework is highly sensitive to content modifications. Even when the extracted QR Code retains a relatively high degree of similarity with the original watermark, any malicious alteration of the document content prevents successful QR-Code decoding and consequently enables reliable integrity attestation and tamper detection.



# CHAPTER 7

## COMPARISON WITH RELATED WORK

---

### 7.1 Image Watermarking

### 7.2 Audio Watermarking

### 7.3 Video Watermarking

### 7.4 Documents Watermarking

---

In this section, a comprehensive comparison of the proposed authenticity and integrity preservation models with representative approaches from the literature is presented. The comparison examines key performance aspects, including imperceptibility, robustness against common benign processing operations and malicious attacks, tamper detection and localization capabilities, computational efficiency, and authentication reliability. For the comparative evaluation, metrics such as PSNR, SSIM, NC, Extraction Rate, BER, Precision, Recall, F1-score, Accuracy, and computational execution time are considered whenever applicable and reported in the corresponding studies.

It should be noted that direct comparisons among watermarking and authentication schemes are inherently challenging due to differences in data-sets, experimental protocols, embedding capacities, attack scenarios, parameter settings, and evaluation methodologies. Furthermore, the reported results correspond to those provided by the respective authors and may represent average, best-case, or data-set specific measurements. The results of the proposed framework are reported as average values over the evaluated data-sets and experimental configurations.

Table 7.1: Comparison of watermark embedding computational time between the proposed model and the SiP-based watermarking technique [1].

| Image      | Size               | SIP [1] Time (s) | Repetitive SiP Time (s) |
|------------|--------------------|------------------|-------------------------|
| people     | $4096 \times 4096$ | 2307.12          | 84.14                   |
| horizon    | $4096 \times 4096$ | 2309.75          | 83.87                   |
| lighthouse | $8192 \times 8192$ | 18636.07         | 513.70                  |
| elephant   | $8192 \times 8192$ | 18297.71         | 504.50                  |

Despite these limitations, the comparative analysis provides a meaningful assessment of the effectiveness, robustness, and practical applicability of the proposed framework relative to existing state-of-the-art approaches.

## 7.1 Image Watermarking

This section presents a comparative evaluation of the proposed image authenticity and integrity watermarking models against representative approaches from the literature. The comparison focuses on key performance aspects, including imperceptibility, watermark extraction reliability, and tamper detection performance. To ensure a consistent assessment, metrics such as PSNR, SSIM, Extraction Rate ( $R_e$ ), Precision, Recall, F1-score, and Accuracy are considered whenever available.

### Repetitive SiP Watermarking Model

Table 7.1 presents the computational-time comparison between the proposed watermarking model and the SiP-based watermarking technique presented in [1]. The experimental results demonstrate that the proposed model significantly reduces the watermark embedding time while preserving comparable imperceptibility characteristics. More specifically, for images of resolution  $4096 \times 4096$ , the proposed model requires approximately 84 seconds for the embedding procedure, whereas the SiP-based methodology requires more than 2300 seconds. Similarly, for images of resolution  $8192 \times 8192$ , the proposed model requires approximately 500 seconds, while the comparative methodology exceeds 18000 seconds.

Table 7.2: Comparison of PSNR and SSIM measurements between the SIP-based watermarking technique [1] and the proposed model.

| Image      | Size               | SIP [1] |      | Repetitive SiP |      |
|------------|--------------------|---------|------|----------------|------|
|            |                    | PSNR    | SSIM | PSNR           | SSIM |
| people     | $4096 \times 4096$ | 33.85   | 0.88 | 33.35          | 0.84 |
| horizon    | $4096 \times 4096$ | 40.79   | 0.97 | 39.28          | 0.97 |
| lighthouse | $8192 \times 8192$ | 34.62   | 0.88 | 34.49          | 0.87 |
| elephant   | $8192 \times 8192$ | 39.23   | 0.97 | 38.42          | 0.96 |

The significant reduction in computational time is mainly attributed to the smaller partitioning *cells* utilized in the proposed model. More specifically, the FFT operations are performed over substantially smaller spatial regions compared to the SIP-based methodology, reducing the computational complexity of the transformation and embedding procedures. For an image of size  $8192 \times 8192$  utilizing a *plane* of size  $7 \times 7$ , the base approach creates *cells* of size  $\approx 1.170 \times 1.170$ , while the repetitive approach utilize an extra *grid* of size  $7 \times 7$ , reduce the *cell* size to  $\approx 167 \times 167$ . Consequently, this *cell* size reduction lowers computationally intensive FFT operations over large regions, significantly accelerating the overall embedding process.

Table 7.2 presents the PSNR and SSIM measurements obtained for the proposed model and the comparative SIP-based watermarking methodology [1]. The experimental results demonstrate that the proposed model preserves high perceptual quality and visual fidelity despite the substantial reduction in computational complexity presented previously. More specifically, the obtained PSNR values remain consistently above 33 dB for all evaluated images, indicating that the visual distortions introduced by the embedding procedure remain imperceptible to the human visual system. Similarly, the SSIM measurements remain very close to 1.0, demonstrating strong structural similarity between the original and the corresponding watermarked images.

Although the comparative SIP-based methodology achieves slightly higher PSNR and SSIM measurements in certain cases, the differences remain relatively small and visually negligible. More specifically, the average reduction in PSNR and SSIM introduced by the proposed model is limited, while the computational-time reduction

Table 7.3: Comparison of PSNR values between the proposed repetitive SiP watermarking model and representative image watermarking approaches from the literature.

| Method                  | Year | Embedding Technique                   | PSNR (dB) |
|-------------------------|------|---------------------------------------|-----------|
| [13]                    | 2013 | Quaternion Fourier Transform          | 24.37     |
| [39]                    | 2017 | Deep Learning Steganography           | 25.21     |
| [40]                    | 2018 | Deep Neural Network Hiding            | 29.08     |
| [41]                    | 2022 | Self-Supervised Latent Space          | 41.79     |
| [28]                    | 2023 | Multi-Bit Watermark + Blockchain      | 41.86     |
| Code-based Multiple SiP | 2023 | FFT Code-Based Multiple SiP Embedding | 41.09     |

remains substantial. Consequently, the proposed methodology achieves a favorable trade-off between computational efficiency and perceptual quality preservation. Furthermore, the experimental results indicate that the proposed model maintains stable imperceptibility characteristics even for very high-resolution images of dimensions up to  $8192 \times 8192$ , demonstrating the scalability and effectiveness of the proposed watermarking methodology for large multimedia objects.

### Code-based Multiple SiP Watermarking Model

Table 7.3 presents a comparison of the PSNR values achieved by the proposed repetitive SiP watermarking model and representative image watermarking approaches from the literature. The results indicate that the proposed model achieves a PSNR value of 41.09 dB, demonstrating a high degree of imperceptibility and preserving the visual quality of the host images after watermark embedding.

More specifically, the obtained PSNR value is significantly higher than those reported by [39], [13], and [40], which achieve PSNR values ranging from 24.37 dB to 29.08 dB. At the same time, the proposed model remains highly competitive with the most recent approaches reported in [41] and [28], exhibiting only a marginal difference of less than 1 dB. Considering that the compared approaches employ different embedding domains and watermarking strategies, these results highlight the effectiveness of the proposed repetitive SiP embedding mechanism in preserving image quality.

Furthermore, the achieved PSNR values are obtained while simultaneously intro-

ducing code-based multiple watermark embedding operations designed to enhance watermark persistence and authentication reliability. Consequently, the proposed methodology demonstrates that the incorporation of multiple watermark structures does not significantly compromise visual fidelity. Overall, the experimental results confirm that the proposed model provides competitive imperceptibility performance while maintaining the robustness, scalability, and authentication capabilities required by the watermarking framework.

### **ROMFAIA Hybrid Watermarking Model**

In this section, a comprehensive comparison of the ROMFAIA system with representative watermarking approaches from the literature is presented. The evaluation focuses on key performance aspects, including imperceptibility, robustness under various common attack scenarios, computational cost, and tamper detection performance in terms of Precision, Recall, F1-score, and Accuracy. It should be noted that variations in experimental setups and image datasets may influence direct comparisons across methods. Furthermore, the reported values are derived from the respective studies and, in most cases, correspond to average results, while the values of the proposed method are computed as averages across the evaluated dataset. Nevertheless, the reported results provide a meaningful indication of the relative performance of the proposed method in relation to existing approaches.

Table 7.4 presents a comparative analysis of the embedding and extraction times of the proposed method against representative approaches from the literature. To ensure a fair comparison, the reported execution times of the proposed method correspond to medium-resolution groups (720p and 1024p), which are comparable to those used in existing works. This allows for a more meaningful assessment of the relative computational cost across different approaches.

Specifically, most of the compared approaches operate on relatively low-resolution images, typically in the range of  $256 \times 256$  to  $512 \times 512$ , and only in limited cases up to  $1024 \times 1024$ . Under these conditions, several methods report lower execution times. For instance, [73] reports embedding and extraction times of 0.27 s and 0.11 s, respectively, while [42] reports 0.70 s and 0.07 s. Comparable results are observed in [74], with execution times of approximately 0.84 s and 0.82 s, whereas higher values are reported in [27] (2.67 s and 2.81 s) and [75] (approximately 4.28 s for embedding and 4.26 s for extraction). These values reflect the reduced computational complexity

Table 7.4: Comparison of computational performance in terms of average embedding ( $T^{Em}$ ) and extraction ( $T^{Ex}$ ) times of the ROMFAIA model.

| Method                    | Image Dimensions                   | $T^{Em}$ (s) | $T^{Ex}$ (s) | Type                    |
|---------------------------|------------------------------------|--------------|--------------|-------------------------|
| Bouarroudj et al. [27]    | $512 \times 512$                   | 2.67         | 2.81         | Robust + Fragile        |
| Singh et al. [73]         | $512 \times 512, 1024 \times 1024$ | 0.27         | 0.11         | Robust + Fragile        |
| Sinhal and Ansari [42]    | $512 \times 512$                   | 0.70         | 0.07         | Robust + Fragile        |
| Singh et al. [74]         | $256 \times 256$                   | 0.84         | 0.82         | Robust                  |
| Khare and Srivastava [75] | $512 \times 512$                   | 4.28         | 4.26         | Robust                  |
| <b>ROMFAIA</b>            | $720p-1024p$                       | <b>3.82</b>  | <b>2.90</b>  | <b>Robust + Fragile</b> |

associated with smaller image sizes.

Table 7.5 presents a comparison of representative watermarking approaches from the literature under non-attacked conditions, in terms of PSNR, SSIM, and BER. In [74], an average PSNR value of 42.99 dB and an SSIM of 0.991 are reported, indicating high imperceptibility, although the method focuses solely on robust watermarking without incorporating fragile integrity verification. Similarly, in [42] and [76], average PSNR values of 40.66 dB and 40.00 dB are reported, respectively, with the former also achieving an average SSIM of 0.941. Higher average PSNR values are observed in [22] and [77], reaching 51.53 dB and 56.31 dB, respectively, while [78] reports an average PSNR of 48.14 dB. On the lower end, [79] reports an average PSNR of 36.58 dB, highlighting the wide range of imperceptibility performance across different approaches, while considering only robust watermarking. Similarly, most reported average SSIM values exceed 0.94, with several approaches achieving values close to 1, such as 0.999 in [22]. In terms of BER, only a subset of the methods explicitly report this metric. For instance, [74] reports an average BER of 0.0030 under standard conditions, while the approaches in [42, 76] achieve an average BER of 0.0000. In contrast, [79] reports a higher average BER value of 0.1106, indicating lower extraction accuracy under the evaluated conditions.

In contrast, the proposed system achieves an average PSNR of 40.81 dB and an average SSIM of 0.971, placing it within the range of several reported approaches while maintaining a high degree of structural preservation. Furthermore, the proposed method achieves an average BER of 0.0000, indicating error-free extraction of

Table 7.5: Comparative imperceptibility results in terms of PSNR, SSIM, and BER under standard (no-attack) conditions of the ROMFAIA model.

| Method                  | PSNR (dB)    | SSIM         | BER           | Type                    |
|-------------------------|--------------|--------------|---------------|-------------------------|
| Singh et al. [74]       | 42.99        | 0.991        | 0.0030        | Robust                  |
| Sinhal and Ansari [42]  | 40.66        | 0.941        | 0.0000        | Robust + Fragile        |
| Sinhal et al. [76]      | 40.00        | –            | 0.0000        | Robust + Fragile        |
| Laishram and Singh [22] | 51.53        | 0.999        | –             | Robust + Fragile        |
| Alshanbari [78]         | 48.14        | –            | –             | Robust + Fragile        |
| Rai and Singh [79]      | 36.58        | –            | 0.1106        | Robust                  |
| Rahman [77]             | 56.31        | –            | –             | Robust + Fragile        |
| <b>ROMFAIA</b>          | <b>40.81</b> | <b>0.971</b> | <b>0.0000</b> | <b>Robust + Fragile</b> |

the embedded information under non-attacked conditions, while supporting a combined robust and fragile watermarking scheme. Finally, the results demonstrate that the proposed method maintains competitive imperceptibility while providing reliable extraction performance and additional integrity verification capabilities, and in several cases achieves higher imperceptibility values compared to existing approaches.

Regarding robustness, Table 7.6 presents a quantitative comparison based on reported BER values of representative approaches under similar attack categories. It can be observed that the BER values reported in the literature vary significantly depending on the type and severity of the applied attack.

Starting with cropping attacks, which directly remove portions of the embedded information, the reported BER values tend to be higher compared to other attack categories. For instance, [22] reports a BER of 0.2266 under a 50% cropping scenario. In comparison, the proposed method achieves a lower BER of 0.1765, where the reported value corresponds to the average performance across multiple cropping directions at 50% severity. This indicates improved resilience under partial data removal, although cropping remains one of the most challenging attack types.

Continuing, filtering attacks produced moderate variations across the examined methods. For Gaussian blur and mean filtering operations, the reported BER values

Table 7.6: Comparison of BER values reported in the literature under common attack scenarios for the ROMFAIA model.

| Attack Type                    | [73]          | [24]   | [26]   | [74]   | [76]          | [22]   | [80]   | [42]   | ROMFAIA       |
|--------------------------------|---------------|--------|--------|--------|---------------|--------|--------|--------|---------------|
| Crop (50%)                     | –             | –      | –      | –      | –             | 0.2266 | –      | –      | <b>0.1765</b> |
| Gaussian Blur ( $3 \times 3$ ) | –             | –      | –      | –      | <b>0.0000</b> | 0.0700 | 0.0143 | 0.0150 | 0.0521        |
| Mean Filter ( $3 \times 3$ )   | <b>0.0012</b> | 0.0705 | –      | –      | 0.0063        | 0.0740 | 0.0988 | 0.0375 | 0.0568        |
| Mean Filter ( $5 \times 5$ )   | <b>0.0076</b> | 0.1371 | –      | –      | –             | –      | –      | –      | 0.0767        |
| Sharpening                     | 0.1731        | –      | –      | 0.1432 | –             | –      | 0.0644 | –      | <b>0.0009</b> |
| Histogram Equalization         | 0.2337        | –      | –      | –      | –             | 0.1582 | –      | –      | <b>0.0033</b> |
| JPEG ( $Q = 90$ )              | –             | –      | 0.0500 | –      | –             | 0.0902 | –      | –      | <b>0.0018</b> |
| JPEG ( $Q = 70$ )              | –             | –      | 0.1200 | –      | <b>0.0000</b> | –      | –      | 0.0225 | 0.0218        |
| JPEG ( $Q = 50$ )              | 0.0796        | –      | 0.2900 | –      | <b>0.0100</b> | 0.2269 | –      | 0.1000 | 0.0532        |
| Gaussian Noise (0.05)          | –             | –      | 0.1500 | –      | –             | –      | –      | –      | <b>0.0003</b> |
| Gaussian Noise (0.10)          | –             | –      | 0.4200 | –      | –             | –      | –      | –      | <b>0.0003</b> |
| Gaussian Noise (0.20)          | –             | –      | 0.5600 | –      | –             | –      | –      | –      | <b>0.0003</b> |
| Scaling (110%)                 | –             | –      | 0.1800 | –      | –             | –      | –      | –      | <b>0.0050</b> |
| Scaling (90%)                  | –             | –      | 0.2600 | –      | –             | –      | –      | –      | <b>0.0121</b> |
| Translation (+5, +5)           | –             | –      | 0.1900 | –      | –             | –      | –      | –      | <b>0.0082</b> |
| Translation (+10, +10)         | –             | –      | 0.4100 | –      | –             | 0.0429 | –      | –      | <b>0.0270</b> |

ranged from near-zero values to approximately 0.14, depending on the attack severity and the underlying methodology [73, 24, 76, 22, 80, 42]. The proposed framework achieved BER values of 0.0521 for Gaussian blur, 0.0568 for mean filtering ( $3 \times 3$ ), and 0.0767 for mean filtering ( $5 \times 5$ ), remaining within the range of values reported in the literature and demonstrating satisfactory robustness against smoothing operations.

For intensity transformations, the proposed method exhibited particularly strong performance. More specifically, BER values of 0.0009 and 0.0033 were obtained for sharpening and histogram equalization, respectively, outperforming several related approaches that reported BER values exceeding 0.15 under the same attack categories [73, 74, 22, 80].

A similar trend was observed for JPEG compression attacks. The proposed framework achieved BER values of 0.0018, 0.0218, and 0.0532 for quality factors  $Q = 90$ ,  $Q = 70$ , and  $Q = 50$ , respectively. These results place the proposed approach among the most robust methods reported in the literature [26, 76, 22, 42, 73], particularly under moderate and high-quality compression settings.

Under Gaussian noise attacks, the proposed method maintained an almost error-



Table 7.7: Comparison of tamper detection performance in terms of Precision, Recall, F1-score, and Accuracy of the ROMFAIA model.

| Method                  | Precision (%) | Recall (%)   | F1-score (%) | Accuracy (%) |
|-------------------------|---------------|--------------|--------------|--------------|
| Meda and Rayachoti [81] | 100.00        | 70.00        | 80.00        | 85.00        |
| Saxena et al. [82]      | 95.62         | 96.83        | 96.22        | –            |
| Singh et al. [73]       | –             | –            | –            | 97.91        |
| Kumaran et al. [84]     | 84.93         | 85.00        | 84.79        | 85.00        |
| Zhang et al. [83]       | 96.28         | 90.50        | 93.30        | 93.50        |
| <b>This Work</b>        | <b>93.64</b>  | <b>94.50</b> | <b>94.06</b> | <b>99.75</b> |

free extraction performance, achieving BER values close to 0.0003 across all evaluated noise levels. This behavior demonstrates substantially improved robustness compared to methods reporting BER values as high as 0.56 under strong noise conditions [26].

Finally, the proposed framework demonstrated strong resistance against geometric distortions. BER values of 0.0050 and 0.0121 were obtained for scaling operations of 110% and 90%, respectively, while translation attacks resulted in BER values of 0.0082 and 0.0270 for shifts of (+5, +5) and (+10, +10). These results indicate improved robustness compared to related approaches [26, 22] and confirm the effectiveness of the proposed watermark representation under spatial transformations.

Table 7.7 presents a comparative evaluation of tamper detection performance in terms of Precision, Recall, F1-score, and Accuracy. To begin with, [81] reports perfect precision (100%) but relatively lower recall (70.00%), resulting in an F1-score of 80.00%, indicating a tendency toward conservative detection. Additionally, [82] achieves the highest F1-score among the compared methods (96.22%), with both precision (95.62%) and recall (96.83%) exceeding 95%. Comparable and well-balanced performance is reported in [83], with an F1-score of 93.30%, while more moderate yet balanced results are observed in [84], with an F1-score of 84.79%. High accuracy (97.91%) is reported in [73], although additional detection metrics are not explicitly provided.

At the same time, ROMFAIA achieves a precision of 93.64%, recall of 94.50%, and F1-score of 94.06%, indicating a well-balanced detection performance. It should be

noted that these values correspond to averages across all evaluated resolution groups. In particular, for higher-resolution images where the *Integrity-plane* is partitioned into a larger number of blocks (i.e., higher  $k$ ), the method enables finer localization of tampered regions, resulting in metric values that approach 100%.

Overall, the results indicate that the proposed method provides comparable performance to existing approaches, while maintaining consistent detection behavior across varying image resolutions and block configurations.

## 7.2 Audio Watermarking

In this section we will present a comparative evaluation of the dual-watermarking approach for the preservation of authenticity of audio files with recent approaches from the literature in regards to auditory imperceptibility and overall payload.

Table 7.8: Comparison of the audio dual-watermarking approach with recent literature approaches.

| Method                     | Year | Domain              | SNR (dB) | Payload (bits/sec) |
|----------------------------|------|---------------------|----------|--------------------|
| Salah et al. [85]          | 2026 | Frequency           | 38.2     | 71.8               |
| Priyadharsini & Kanhe [86] | 2026 | Frequency           | 61–78.6  | 1200               |
| Our Model                  | 2022 | Spatial + Frequency | 64.09    | 2184.5–4369.1      |

Table 7.8 compares our audio dual-watermarking approach against two recent frequency-domain methods. Salah et al. [85] report an SNR of 38.2 dB with a payload of 71.8 bits/sec, a conservative embedding profile that trades capacity for a simpler, lower-complexity frequency-domain scheme. Priyadharsini and Kanhe [86] achieve a considerably higher SNR range (61–78.6 dB) alongside a payload of 1200 bits/sec, reflecting an aggressive frequency-domain embedding strategy optimized primarily for capacity.

Our model, although published earlier (2022), attains an SNR of 64.09 dB — comfortably within the upper range reported by Priyadharsini and Kanhe [86] and substantially above Salah et al. [85] — while delivering a substantially higher payload of 2184.5–4369.1 bits/sec. Unlike the two competing methods, which operate exclusively in the frequency domain, our approach embeds a fixed  $128 \times 128$  grayscale

watermark image into the host audio using a combined spatial-frequency scheme, corresponding to a total embedding capacity of  $128 \times 128 \times 8 = 131,072$  bits per audio clip (8 bits per pixel). Since this fixed payload is embedded once per clip, the effective bits/sec rate is inversely proportional to clip duration: for our test set of audio signals ranging from 30 seconds to 1 minute, this yields an effective payload of  $131072/60 \approx 2184.5$  bits/sec for the longest clips and  $131072/30 \approx 4369.1$  bits/sec for the shortest, precisely matching the reported range.

This distinction is important when comparing against [86] and [85], whose payload figures are considerably lower and reported as single fixed values, presumably under a fixed-duration test protocol. Our model’s payload exceeds both competing methods by roughly one to two orders of magnitude, while still maintaining a competitive SNR — indicating that the spatial-frequency hybrid embedding is substantially more capacity-efficient than a purely frequency-domain approach, at comparable imperceptibility levels. This is best explained by the fact that our scheme embeds a full grayscale image (with per-pixel intensity information) rather than a compact binary sequence, giving it a structurally higher capacity ceiling.

Finally, it is notable that both competing methods post-date our model by four years, yet our approach not only remains competitive in SNR but substantially outperforms both in payload capacity despite testing across a range of clip durations rather than a single fixed one. This suggests the spatial-frequency dual-embedding principle is a highly capacity-efficient architectural choice for practical audio watermarking scenarios where full-image-scale, grayscale watermark recovery is required.

### 7.3 Video Watermarking

In this section we will present a comparative evaluation of the proposed video authenticity and integrity watermarking models against representative video approaches from the state-of-the-art. The comparison focuses on run-time performance, imperceptibility and tamper detection performance. To ensure a consistent assessment we will use PSNR, SSIM, BER, Precision, Recall, F1-score, and Accuracy whenever available.

Table 7.9: Comparative performance (run-time) results with state-of-the-art methods including both traditional transform domain techniques and also Deep-Learning techniques, at resolution of  $1280 \times 720$  for the code-based hash-chain video watermarking model.

| Presented in:               | Technique           | Resolution        | (sec./frame) | Year |
|-----------------------------|---------------------|-------------------|--------------|------|
| [87]                        | DT-CWT              | $1280 \times 720$ | 9.6          | 2022 |
| [88]                        | Deep Learning (GPU) | $1280 \times 720$ | 2.5          | 2023 |
| [89]                        | Deep Learning (GPU) | $1280 \times 720$ | 2.3          | 2024 |
| [6]                         | DCT                 | $1280 \times 720$ | 12.0         | 2024 |
| [90]                        | JND-DCT             | $1280 \times 720$ | 7.8          | 2025 |
| [91]                        | FSCT                | $1280 \times 720$ | 8.2          | 2025 |
| Code-based hash-chain model | FFT/HASH-CHAIN      | $1280 \times 720$ | 2.41         | 2025 |

### Code-based Hash Chain Video Watermarking Model

Regarding performance, Table 7.9 presents a comparative analysis with representative state-of-the-art approaches at a resolution of  $1280 \times 720$ , where reported values are taken directly from the corresponding publications. While deep learning based methods typically achieve lower per-frame run-time, they often rely on specialized hardware and incur additional training and deployment overhead. In contrast, traditional transform-domain methods exhibit significantly higher computational complexity. The proposed scheme offers a favorable trade-off by outperforming classical transform-based approaches while maintaining competitive run-time relative to learning-based methods.

Table 7.10 presents a comparative evaluation of the proposed watermarking method against representative approaches from the literature using PSNR and SSIM as quality assessment metrics. These metrics jointly evaluate pixel-level fidelity and perceptual structural preservation, both of which are essential for watermarking systems designed for authenticity verification and tamper detection.

The results indicate that the proposed method achieves a balanced combination of high PSNR and SSIM values, demonstrating its ability to preserve the visual quality of the host content while embedding authentication information. Although several

Table 7.10: Comparative imperceptibility results showcasing PSNR and SSIM in contrast to other approaches for the code-based hash-chain video watermarking model.

| Metric | [93] | [90]  | [89]  | [94]  | [95]  | [96]  | [88]  | [92]  | Code-based hash-chain model |
|--------|------|-------|-------|-------|-------|-------|-------|-------|-----------------------------|
| PSNR   | 37.5 | 44.77 | 34.79 | 35.34 | 27.46 | 36.2  | 37    | 46.54 | 35.04                       |
| SSIM   | –    | 0.957 | 0.949 | 0.954 | 0.903 | 0.961 | 0.985 | 0.98  | 1.000                       |

Table 7.11: Quantitative comparison based on reported BER values of representative video watermarking approaches under similar attack categories for the code-based hash-chain video watermarking model.

| Presented in                | Technique                  | Gaussian Blur<br>(7×7) | Gaussian Noise<br>( $\sigma=0.04$ ) | Gaussian Noise<br>( $\sigma=0.05$ ) | H.264 Compression<br>(CRF=28) | H.264 Compression<br>(CRF=22) | JPEG Compression<br>(Q=70) | Frame Drop<br>(20/30/50 %) | Frame Swap<br>(20/30/50 %) |
|-----------------------------|----------------------------|------------------------|-------------------------------------|-------------------------------------|-------------------------------|-------------------------------|----------------------------|----------------------------|----------------------------|
| [96]                        | DWT (2013)                 | 0.0932                 | 0.0099                              | 0.0139                              | –                             | 0.1071                        | –                          | (–/–/0.2083)               | (–/–/0.1104)               |
| [97]                        | FFT Fourier Moments (2016) | –                      | –                                   | –                                   | 0.5457                        | –                             | 0                          | –                          | –                          |
| [94]                        | DL (2018)                  | 0.3259                 | 0.0873                              | 0.0925                              | –                             | 0.2015                        | –                          | (–/–/0.2475)               | (–/–/0.0086)               |
| [98]                        | FFT Fourier Moments (2020) | –                      | –                                   | –                                   | 0.0080                        | –                             | 0                          | –                          | –                          |
| [99]                        | T-QR (2022)                | 0.0177                 | –                                   | 0.0278                              | –                             | –                             | –                          | (–/0.0006/0.0006)          | (–/0.0006/–)               |
| [92]                        | Zernike Moments (2022)     | –                      | –                                   | –                                   | 0.0026                        | –                             | –                          | (0.0051/0.0053/–)          | –                          |
| [93]                        | DL (2023)                  | –                      | 0.0003                              | –                                   | –                             | 0.041                         | –                          | (–/–/0.0019)               | (–/–/0.0011)               |
| [88]                        | DL (2023)                  | 0.0191                 | 0.0144                              | 0.0250                              | –                             | 0.0706                        | –                          | (–/–/0.0101)               | (–/–/0.0065)               |
| [89]                        | DCT-DL (2024)              | [0.0001–0.0266]        | –                                   | [0.0000–0.0134]                     | [0.0001–0.0244]               | –                             | –                          | (–/–/[0.0000–0.0241])      | (–/–/[0.0000–0.0252])      |
| [100]                       | DWT-DL (2024)              | –                      | –                                   | –                                   | –                             | –                             | –                          | (0.05/0.074/–)             | –                          |
| [101]                       | DL (2025)                  | –                      | –                                   | –                                   | –                             | –                             | 0.025 ± 0.011              | –                          | –                          |
| [90]                        | JND-DCT (2025)             | –                      | –                                   | 0.14                                | 0.18                          | –                             | –                          | –                          | –                          |
| [102]                       | DL (2026)                  | –                      | –                                   | –                                   | –                             | –                             | –                          | (0.024/0.034/–)            | –                          |
| Code-based hash-chain model | FFT + Hash Chain           | 0.0427                 | 0.003                               | 0.003                               | 0.11                          | 0.08                          | 0.025                      | (0.0002/0.003/0.003)       | (0.0002/0.002/0.002)       |

approaches, particularly [90] and [92], report slightly higher PSNR values reaching up to 46.54 dB, the proposed method remains highly competitive while maintaining excellent structural similarity. High PSNR values indicate limited pixel-wise distortion; however, they do not necessarily guarantee the preservation of perceptually important structural information.

In contrast, SSIM provides a more representative assessment of visual quality by evaluating luminance, contrast, and structural consistency. The proposed method achieves SSIM values close to unity, indicating that the embedded watermark introduces negligible perceptual degradation and preserves the structural characteristics of the original content. This is particularly important in authenticity-preservation applications, where both visual transparency and reliable watermark extraction must be simultaneously maintained.

Table 7.11 presents a comparative evaluation of watermark robustness under common video processing attacks using BER as the primary performance metric. The results reveal notable differences among existing approaches, reflecting distinct em-

Table 7.12: Comparison in tamper detection with similar approaches (in chronological order) for the code-based hash-chain video watermarking model.

| Presented in:               | Precision | Recall | Accuracy | F1-Score |
|-----------------------------|-----------|--------|----------|----------|
| [103]                       | 75.12     | 87.07  | –        | 80.69    |
| [104]                       | –         | –      | 98.00    | 99.00    |
| [105]                       | 93.78     | 93.90  | 94.89    | 93.00    |
| Code-based hash-chain model | 100.0     | 100.0  | 100.0    | 100.0    |

bedding strategies and robustness objectives.

For Gaussian blur and Gaussian noise attacks, the proposed method exhibits stable performance, achieving BER values comparable to several state-of-the-art approaches. In particular, methods such as [99] report very low BER values under blur and noise distortions, while the proposed method maintains consistent watermark recovery with BER values of 0.0427 under Gaussian blur and 0.003 under Gaussian noise ( $\sigma = 0.04$  and  $\sigma = 0.05$ ).

Under H.264 compression, the proposed approach achieves BER values of 0.11 (CRF = 28) and 0.08 (CRF = 22), preserving watermark detectability under practical compression settings. Although methods such as [92] and [98] report lower BER values under specific compression configurations, the proposed method maintains competitive performance while simultaneously addressing a broader range of attack scenarios.

The most significant advantage of the proposed method is observed under temporal attacks. While several approaches, including [96], [94], exhibit increasing BER as the frame dropping ratio increases, the proposed method maintains BER values on the order of  $10^{-4}$  to  $10^{-3}$  for both Frame Dropping and Frame Swapping attacks. This robustness is attributed to the frame-independent embedding strategy, where watermark information is embedded and extracted independently in each frame without relying on temporal continuity. Consequently, frame removal or reordering has minimal impact on watermark recovery. Overall, the results demonstrate that the proposed method achieves competitive robustness against spatial and compression attacks while exhibiting particularly strong resistance to temporal distortions.

In Table 7.12, a comparison of the integrity-preservation capabilities of the proposed scheme and related approaches is presented. Unlike existing methods that rely on statistical watermark properties or probabilistic detection mechanisms, which may introduce false positives or false negatives, the proposed scheme adopts a deterministic hash-based integrity verification strategy. More specifically, the cryptographic hash of all video frames is embedded within the audio channel, enabling direct frame-level integrity validation during the verification process. As a result, any modification to a frame produces a hash mismatch, allowing tampering to be detected with certainty and eliminating the uncertainty associated with probabilistic approaches.

This deterministic design provides reliable tamper detection while maintaining the high visual quality demonstrated in Table 7.10. Although several existing methods achieve competitive PSNR and SSIM values, they generally rely on similarity measures or statistical thresholds for authenticity verification, which may still lead to detection errors. In contrast, the proposed scheme combines high perceptual quality with deterministic integrity verification, making it particularly suitable for security-critical applications such as surveillance systems, digital forensics, legal evidence management, and medical imaging.

### **Cross-Frame Permutation Based Video Watermarking Model**

The comparative evaluation of tamper detection approaches is commonly based on statistical metrics such as Precision, Recall, Accuracy, and F1-Score. These metrics are particularly relevant for learning-based and probabilistic detection frameworks, where classification decisions are subject to uncertainty. In contrast, the proposed method employs a deterministic hash-based verification mechanism embedded at the LSB level. Since any modification to the protected content alters the corresponding hash value, tampering can be detected with certainty, assuming collision-resistant hashing and reliable extraction.

Table 7.13 presents a comparison of the proposed approach with representative video forgery detection methods from the literature. Most existing approaches achieve high, yet imperfect, performance, with Precision values ranging from 69.63% to 100%, Recall values from 80.42% to 100%, Accuracy values from 87.25% to 99.5%, and F1-Scores from 80.69% to 99.4%. Among the compared methods, [106] reports the strongest overall performance, achieving 100% Precision, 99% Recall, 99.5% Accuracy, and a 99.4% F1-Score. Similarly, [107] achieves perfect Recall, while [108] and [104]

Table 7.13: Performance comparison with representative video tampering detection and localization approaches based on Precision, Recall, Accuracy, and F1-Score of the cross-frame permutation based video watermarking model. The metrics are averaged across all attack scenarios.

| Method                                     | Technique                                      | Precision    | Recall       | Accuracy     | F1-Score     |
|--------------------------------------------|------------------------------------------------|--------------|--------------|--------------|--------------|
| [109]                                      | Compressive Sensing                            | 89.9         | 90.5         | 89.6         | –            |
| [108]                                      | Deep Learning                                  | 97.31        | 91.05        | 98.45        | 94.07        |
| [110]                                      | Inter-frame Analysis                           | 87.97        | 91.42        | 92.84        | 91.89        |
| [107]                                      | Similarity Analysis                            | 98.07        | 100          | 99.01        | –            |
| [106]                                      | Correlation-Based                              | 100          | 99           | 99.5         | 99.4         |
| [111]                                      | Haralick Features                              | 86           | 96           | –            | 91           |
| [112]                                      | HEVC Compressed Domain                         | 90.77        | 93.4         | 87.25        | 89.68        |
| [113]                                      | CNN-Based                                      | 69.63        | 80.42        | 91           | 88.35        |
| [103]                                      | Sequential + Patch Analysis                    | 75.117       | 87.07        | –            | 80.69        |
| [104]                                      | Deep Learning CMFD                             | 93.78        | 93.9         | 94.89        | 93           |
| [105]                                      | Adaptive Segmentation + DNN                    | 93.78        | 93.9         | 94.89        | 94           |
| <b>Cross-frame permutation based model</b> | <b>Cross-Frame Permutation + SHA-256 + LSB</b> | <b>99.73</b> | <b>99.27</b> | <b>98.67</b> | <b>99.23</b> |

demonstrate consistently high performance across all reported metrics.

The proposed cross-frame permutation based model achieves an average Accuracy of 99.73%, Precision of 99.27%, Recall of 98.67%, and F1-Score of 99.23% across all evaluated single-frame and multi-frame tampering scenarios. These results outperform the majority of the compared approaches and remain highly competitive with the best-performing methods reported in the literature. In particular, the proposed framework consistently maintains high detection and localization performance regardless of the attack type, demonstrating its robustness under both spatial and temporal tampering conditions. The obtained results confirm the effectiveness of the proposed watermarking framework and highlight its suitability for authenticity-preservation applications requiring reliable video forgery detection and localization.

## 7.4 Documents Watermarking

In this section we will present a comparative evaluation of the proposed PDF authenticity and integrity watermarking model against representative document approaches from the state-of-the-art. The comparison focuses on imperceptibility of the water-



Table 7.14: Comparison of the exhibited PSNR, SSIM and MSE values in regards to quality and imperceptibility of PDF pages with the results exhibited by related approaches.

| Presented in                             | PSNR          | SSIM  | MSE   |
|------------------------------------------|---------------|-------|-------|
| [114]                                    | [63.15-70.88] | -     | -     |
| [17]                                     | 74.6          | 0.893 | 39.43 |
| [18]                                     | 29.975        | 0.87  | -     |
| [115]                                    | 44.46         | 0.996 | 2.35  |
| [116]                                    | 40.10         | 0.962 | 0.07  |
| [19]                                     | 28.64         | -     | -     |
| [117]                                    | 59.96         | 0.99  | 0.07  |
| [118]                                    | [36-55]       | -     | -     |
| [119]                                    | 55.68         | 0.99  | 2.81  |
| PDF Watermarking Model utilizing QR-Code | 57.59         | 0.999 | 0.113 |

mark on the PDF pages. To ensure a consistent assessment we will use PSNR, MSE and SSIM, whenever available.

Table 7.14 presents a comparison between the proposed PDF watermarking scheme and representative approaches from the literature using PSNR, SSIM, and MSE as evaluation metrics. The proposed method achieves a PSNR value of 57.59 dB, demonstrating high visual quality and low embedding distortion. This value is comparable to, and in several cases exceeds, the PSNR values reported by many existing approaches. Although [17] reports a higher PSNR value of 74.6 dB and [114] reports PSNR values ranging from 63.15 dB to 70.88 dB, these studies provide limited complementary quality metrics, making a comprehensive assessment of visual fidelity and robustness more difficult.

Regarding structural similarity, the proposed scheme achieves an SSIM value of 0.999, indicating an almost perfect preservation of the visual and structural characteristics of the original PDF document. This result surpasses those reported by several related works, including [17] (SSIM = 0.893), [18] (SSIM = 0.87), and [119]

(SSIM = 0.99). The obtained SSIM value confirms that the watermark embedding process introduces negligible perceptual distortion while maintaining the integrity of the document content.

Although MSE values are not explicitly reported for the proposed method, the combination of high PSNR and near-unity SSIM values indicates a very low level of distortion. While some approaches report lower MSE values, such as [18] (MSE = 2.35) and [116] (MSE = 0.07), direct comparisons should be interpreted cautiously due to differences in datasets, embedding capacities, and experimental configurations. Overall, the results demonstrate that the proposed watermarking scheme achieves an effective balance between imperceptibility, structural preservation, and watermark embedding performance, making it highly competitive with existing PDF watermarking approaches.

# CHAPTER 8

## PERFORMANCE ANALYSIS

---

### 8.1 Authenticity and Integrity Preservation

### 8.2 Practical Feasibility of the Proposed Models

### 8.3 Limitations and Challenges

---

## 8.1 Authenticity and Integrity Preservation

In this section, the performance of the proposed models is analyzed based on the experimental results presented throughout the dissertation. More specifically, the discussion focuses on the effectiveness of the proposed methodologies in preserving the authenticity and integrity of digital objects, as well as their robustness against malicious manipulations and common attack scenarios. Furthermore, the computational performance, scalability, and practical applicability of the proposed approaches are examined in order to assess their suitability for real-world deployment.

### 8.1.1 Image

The experimental results demonstrate that the proposed image watermarking models successfully achieve the primary objectives of authenticity and integrity preservation while maintaining high visual quality. More specifically, the obtained PSNR values consistently remained above 33 dB, while the corresponding SSIM values approached unity, indicating that the embedding procedures introduce only limited perceptual distortion and preserve the structural characteristics of the protected images.

The repetitive SiP watermarking model highlights the effectiveness of redundancy as a mechanism for improving robustness against cropping and partial-content attacks. By distributing watermark information across multiple image regions, the proposed model significantly increases the probability of successful watermark recovery even when substantial portions of the image are removed. Despite embedding multiple watermark replicas, the model preserves high visual quality, achieving PSNR values between 33.35 dB and 39.28 dB and SSIM values ranging from 0.84 to 0.97. Furthermore, compared to the baseline SiP watermarking approach, the repetitive embedding strategy enables successful watermark extraction even under severe cropping conditions while simultaneously reducing the computational complexity of the embedding procedure through the utilization of smaller transform regions.

The code-based multiple SiP watermarking model further extends the representation capabilities of traditional SiP-based approaches by encoding multiple integers into a single watermark code. This design increases watermark capacity, uniqueness, and authentication capability while preserving imperceptibility. The experimental evaluation reports an average PSNR of 41.09 dB and an average SSIM of 0.952, while achieving error-free watermark extraction under no-attack conditions with an average BER of 0.0000. Comparative evaluation further demonstrates that the proposed framework remains highly competitive with state-of-the-art methods despite supporting substantially richer watermark representations and stronger authentication capabilities.

The proposed cross-block integrity-preservation model demonstrated highly accurate tamper detection and localization performance through deterministic cryptographic verification relationships established among image blocks. The obtained Precision, Recall, F1-score, and Accuracy values approached unity, confirming the effectiveness of the proposed permutation-based verification mechanism. Unlike learning-based approaches, the proposed framework does not require training data and provides deterministic, explainable, and reliable integrity verification decisions. Furthermore, the introduced  $\Lambda$ -relations establish interconnected verification dependencies among image blocks, reducing false-positive detections while improving localization accuracy.

Building upon these contributions, the proposed ROMFAIA framework successfully integrates robust authenticity preservation and fragile integrity verification into a unified protection framework for forensic medical images. The sequential deployment

of the two watermarking layers preserves high visual quality, achieving an average PSNR of 40.81 dB and an average SSIM of 0.971, while maintaining an Extraction-Rate of 100% and a BER equal to 0 under non-attacked conditions. The authenticity component demonstrates strong robustness against cropping, filtering, JPEG compression, and geometric transformations, maintaining high extraction rates even under challenging attack scenarios. At the same time, the fragile integrity component consistently detects and accurately localizes tampered regions across all evaluated attacks without producing false-positive detections. An additional improvement introduced in ROM-FAIA is the replacement of the previously deployed  $SHA_{256}$  verification mechanism with HMAC-SHA-256, incorporating secret-key authentication into the integrity verification procedure. This enhancement prevents unauthorized regeneration of valid verification information, substantially strengthening the resistance of the framework against forgery and malicious re-embedding attacks.

Overall, the obtained results demonstrate that the proposed image watermarking frameworks achieve an effective balance among imperceptibility, robustness, authentication capability, and integrity preservation. Collectively, the developed methodologies provide scalable, computationally efficient, and highly reliable solutions for authenticity verification, tamper detection, and integrity preservation, making them suitable for both general-purpose digital images and high-security medical imaging applications.

### 8.1.2 Audio

The experimental results demonstrate that the proposed audio authenticity and integrity preservation models successfully achieve their design objectives while maintaining high perceptual quality. More specifically, the dual-layer audio watermarking model achieved average PSNR values of 65.09 dB under noise attacks and 63.09 dB under down-sampling attacks, indicating that the embedding procedure introduces negligible audible distortion despite the relatively high embedding payload of 131,072 bits. These results confirm that the deployment of LSB-based embedding preserves the acoustic fidelity of the protected audio signals while providing sufficient capacity for authenticity-preservation information.

The robustness evaluation further demonstrates the effectiveness of the proposed dual-layer watermarking architecture. Under noise injection and down-sampling attacks, the proposed model achieved a 100% watermark extraction success rate, while

maintaining high perceptual quality. Furthermore, successful extraction of both the watermarked image and the embedded watermark remained possible for all cropping attacks up to 50% of the audio content. Even under severe cropping attacks affecting 75% of the signal, the embedded watermark could still be recovered in specific cases due to the reconstruction capabilities provided by the underlying SiP representation. These results highlight the benefits of utilizing a watermarked image as an intermediate carrier, since the embedded authenticity information remains recoverable even when the carrier itself is partially degraded.

Regarding integrity preservation, the permutation-based cross-audio block model demonstrated reliable tamper detection capabilities across all evaluated attack categories. More specifically, large-scale cropping, localized cropping, amplification attacks, and combinations of signal-processing modifications were successfully detected and localized. The observed performance can be attributed to the XOR-based integrity relations established between audio blocks and the utilization of HMAC-derived verification information. Any modification affecting the protected audio content disrupts the corresponding verification relations, enabling deterministic tamper detection without relying on statistical thresholds or learning-based classification models.

A different behavior was observed under resampling attacks. Although the proposed framework successfully detected the integrity violation, accurate localization was not possible because the resampling process alters the sample positions and consequently modifies the structure of the generated audio blocks. Nevertheless, the ability to consistently identify the presence of tampering demonstrates the resilience of the proposed verification mechanism even under severe structural modifications of the audio signal.

Overall, the obtained results indicate that the proposed audio models achieve a favorable balance among imperceptibility, robustness, authenticity verification, and integrity preservation. The experimental evaluation confirms their suitability for applications requiring high audio quality, reliable authenticity attestation, and deterministic tamper detection and localization capabilities.

### **8.1.3 Video**

The experimental results demonstrate that the proposed video authenticity and integrity preservation models successfully achieve their design objectives while main-

taining high visual quality. More specifically, both models preserve the perceptual characteristics of the protected video content while providing reliable authenticity verification and tamper detection capabilities under a wide range of signal-processing, geometric, temporal, and content-manipulation attacks.

The code-based hash-chain video watermarking model achieves high imperceptibility across all evaluated resolutions, codecs, and frame rates. The obtained PSNR values range from 32.31 dB to 38.73 dB, while SSIM remains equal to 1.0 in all evaluated configurations, indicating that the embedding process introduces negligible visual distortion. These results demonstrate that the proposed frequency-domain embedding strategy successfully preserves the visual characteristics of the protected videos regardless of the underlying codec or temporal sampling configuration. Furthermore, the model achieves high robustness against the majority of evaluated attacks. Extraction rates remain close to 100% for noise attacks, histogram equalization, mild gamma correction, compression, and moderate cropping operations, while even under severe attacks such as large-kernel blurring and aggressive compression, extraction rates remain above 89% with BER values below 0.05. The obtained results indicate that the proposed framework achieves a favorable balance between imperceptibility and robustness while preserving reliable authenticity verification capabilities.

The robustness evaluation additionally demonstrates that the proposed global frame–audio hash-chain mechanism effectively strengthens integrity verification. Unlike conventional approaches that verify individual frames independently, the proposed framework derives a cryptographic representation from the complete video sequence and embeds it into the audio stream. Consequently, modifications affecting either the visual or auditory components alter the final integrity representation, enabling reliable detection of spatial, temporal, and cross-modal manipulations. The consistently high extraction rates and near-zero BER values observed across the evaluated attack scenarios verify the effectiveness of this design.

Regarding integrity preservation, the cross-frame permutation-based video model achieved excellent detection and localization performance across all evaluated attack categories. For single-frame attack scenarios, the proposed framework achieved Accuracy values between 0.994 and 1.000, Precision values between 0.970 and 1.000, Recall values up to 1.000, and F1-scores ranging from 0.984 to 1.000. Similarly, under multi-frame attack scenarios, Accuracy remained between 0.990 and 0.9999, while F1-scores remained above 0.989 for all evaluated attacks. These results demonstrate

the ability of the proposed framework to reliably detect and localize object insertion, object deletion, copy-move manipulations, and combined attack scenarios. The observed performance can be attributed to the introduced cross-frame and cross-block verification relationships, which establish integrity dependencies not only within individual frames but also across different temporal locations of the video sequence. Furthermore, the additional binding of verification information to the audio stream strengthens the integrity guarantees of the framework by introducing a supplementary verification layer spanning multiple modalities. Consequently, modifications affecting either the visual content or the associated audio stream disrupt the verification relationships, enabling reliable detection and localization of malicious manipulations.

The perceptual quality measurements further demonstrate the suitability of the proposed integrity-preservation framework for practical applications. More specifically, the achieved PSNR values range from 41.23 dB to 50.57 dB, with an average value of 46.11 dB, while the average SSIM reaches 0.987. These measurements indicate that the relatively high integrity payload introduced by the framework does not significantly affect the visual quality of the protected videos. Consequently, the proposed model successfully combines high imperceptibility with reliable tamper detection and localization capabilities.

Overall, the obtained results indicate that the proposed video watermarking models achieve a favorable balance among imperceptibility, robustness, authenticity verification, and integrity preservation. The experimental evaluation confirms their suitability for applications such as digital forensics, surveillance systems, multimedia authentication, and evidential video protection, where reliable verification of both authenticity and integrity is of critical importance.

#### **8.1.4 Document**

The experimental results demonstrate that the proposed PDF watermarking model successfully preserves authenticity and integrity while maintaining the visual appearance and functionality of the protected documents. More specifically, the obtained imperceptibility measurements indicate that the embedding process introduces negligible visual distortion. The evaluated PDF pages achieved an average PSNR of 57.60 dB and an average SSIM of 0.99986, confirming that the text-color modifications employed for watermark embedding remain visually imperceptible and do not affect



document readability or usability. Furthermore, the extracted QR Codes remained fully functional for all evaluated pages, enabling successful authenticity and integrity attestation.

The robustness evaluation further demonstrates the effectiveness of the proposed authenticity-preservation mechanism under common PDF processing operations. More specifically, operations such as markup insertion, digital signatures, page rotation, metadata deletion, margin modifications, scaling, compression, and page cropping preserved the functionality of the extracted QR Code and achieved successful authenticity verification in all evaluated scenarios. Even under aggressive cropping, scaling, and compression attacks, the extracted QR Code maintained BER values equal to zero while remaining fully decodable. These results indicate that the proposed framework can preserve authenticity information during routine document editing, storage, transmission, and printing procedures.

Regarding integrity preservation, the proposed framework exhibited high sensitivity to malicious document modifications. More specifically, PDF-to-image conversion, text insertion, text deletion, and text modification attacks consistently disrupted the embedded watermark information, resulting in non-functional QR Codes and successful tamper detection. Depending on the position and extent of the modification, BER values increased up to 0.6556, while SSIM values decreased to as low as 0.0000, indicating severe degradation of the extracted watermark. Even in scenarios where the extracted QR Code retained partial similarity with the original watermark, successful decoding was not possible, allowing reliable integrity attestation and tamper detection.

The obtained results can be attributed to the combined utilization of QR-Code watermarking, Arnold transformations, and the additional permutation layer introduced by the proposed framework. The deployment of the permutation stage conceals the characteristic structural patterns of the QR Code, preventing attackers from identifying potential watermark locations, while the text-color embedding strategy avoids document rasterization and preserves the original PDF structure. Consequently, the proposed methodology successfully combines high imperceptibility, robust authenticity preservation, and reliable integrity attestation while maintaining the functionality, readability, and visual quality of the protected documents.

## 8.2 Practical Feasibility of the Proposed Models

The experimental evaluation regarding the run-time demonstrates that the proposed authenticity and integrity preservation models exhibit practical computational requirements and favorable scalability characteristics across all considered digital object categories. As expected, the computational complexity increases with the amount of information processed, ranging from images and audio signals to videos and multi-page PDF documents. Nevertheless, the obtained results indicate that the proposed methodologies maintain predictable resource requirements and can be deployed in practical authenticity and integrity preservation applications.

For image watermarking, the repetitive SiP model demonstrated a substantial improvement in computational efficiency compared to the original SiP-based framework. More specifically, the required embedding time was reduced from approximately 2,300 seconds to 84 seconds for images of resolution  $4096 \times 4096$ , while for images of resolution  $8192 \times 8192$  the execution time decreased from more than 18,000 seconds to approximately 500 seconds. These results verify that the proposed grid-based embedding strategy significantly improves scalability while preserving the robustness benefits of repetitive watermark embedding.

Building upon these improvements, the ROMFAIA framework also demonstrated favorable computational performance despite integrating both robust authenticity preservation and fragile integrity verification within a single framework. The average embedding and extraction times across all evaluated resolution groups were 11.44 and 9.67 seconds, respectively. Even for 4K images, the complete embedding and extraction procedures required only 39.91 and 35.32 seconds, respectively, demonstrating the practical applicability of the framework for high-resolution medical images. Furthermore, the experimental evaluation showed that the authenticity-related processing constitutes the dominant portion of the execution time due to the deployment of frequency-domain watermark embedding, whereas the integrity-preservation procedures remain computationally lightweight, relying primarily on cryptographic hash computations and LSB modifications. Overall, the observed execution times increase predictably with image resolution, confirming the scalability of the proposed framework.

The proposed audio watermarking and integrity-preservation models exhibited the lowest computational requirements among all evaluated models. This behavior

can be attributed to the utilization of lightweight LSB embedding operations and efficient cryptographic verification mechanisms that operate directly on audio samples. Consequently, the proposed audio frameworks remain suitable for deployment in real-time and near real-time environments where computational efficiency is of particular importance.

Video processing naturally introduces higher computational requirements due to the large volume of visual and temporal information that must be processed. Nevertheless, the scalability analysis demonstrated predictable growth in execution time as video resolution increased. More specifically, the average processing time per frame increased from approximately 3.07 seconds for 720p videos to 5.93 seconds for 1080p videos, 10.52 seconds for 2K videos, and 25.03 seconds for 4K videos. Despite this increase, the observed behavior remained approximately linear with respect to the amount of processed visual information. Furthermore, stable performance was observed across different codecs and frame-rate configurations, indicating that the proposed framework can be effectively deployed under a wide range of practical video-processing scenarios.

The PDF watermarking framework also exhibited stable and predictable computational behavior. The embedding time increased from approximately 20 seconds for single-page documents to slightly above 300 seconds for documents consisting of sixteen pages. At the same time, CPU utilization remained moderate and memory consumption increased proportionally with document size. Additionally, the number of Arnold transformation iterations introduced only minor variations in execution time and resource utilization, demonstrating that the employed transformations do not constitute a significant computational bottleneck.

Overall, the obtained results indicate that the computational requirements of the proposed methodologies remain proportional to the complexity of the protected digital objects. The observed execution times, resource utilization measurements, and scalability characteristics confirm the practical feasibility of the proposed frameworks for authenticity and integrity preservation applications. Furthermore, the deterministic nature of the proposed verification mechanisms eliminates the training and model-maintenance costs commonly associated with machine-learning-based approaches, further improving their suitability for real-world deployment.

### 8.3 Limitations and Challenges

#### Minimum Embedding Cell Size Constraints

A common limitation of the proposed authenticity-preservation frameworks based on SiPs originates from the geometric requirements of the annulus-based embedding mechanism. More specifically, watermark embedding relies on the comparison of the average values contained in the red and blue concentric annular regions constructed within each SiP cell. Consequently, the embedding cells cannot be reduced arbitrarily, since excessively small cells may not provide sufficient pixels for the formation of stable annular structures.

Let  $G \times G$  denote the dimensions of an embedding cell. Since each cell is a square region, the horizontal and vertical radii of the outer (red) annulus are identical and are given by

$$R_{r_x} = R_{r_y} = R_r = \left\lfloor \frac{G}{2} \right\rfloor. \quad (8.1)$$

Similarly, the inner (blue) annulus is also circular and therefore

$$R_{b_x} = R_{b_y} = R_b = R_r - p_r, \quad (8.2)$$

where  $p_r$  denotes the width of the red annulus. To guarantee the existence of both annular regions, the radius of the blue annulus must be at least equal to its width  $p_b$ , yielding

$$R_b \geq p_b. \quad (8.3)$$

Substituting the definition of  $R_b$  gives

$$R_r \geq p_r + p_b. \quad (8.4)$$

Assuming the parameter values employed throughout this dissertation ( $p_r = p_b = 2$ ), the above condition becomes

$$R_r \geq 4. \quad (8.5)$$

Since

$$R_r = \left\lfloor \frac{G}{2} \right\rfloor, \quad (8.6)$$

the minimum admissible cell dimension is

$$G_{\min} = 8. \quad (8.7)$$

Consequently, the smallest embedding cell that guarantees the existence of both annular regions is an  $8 \times 8$  pixel square.

Although watermark extraction remains possible for relatively small embedding cells, reducing the cell dimensions decreases the number of available pixels and consequently the statistical diversity among neighboring cells. As a result, the differences between annulus-based measurements become less pronounced, increasing the probability of ambiguous comparisons during watermark extraction. Therefore, the minimum embedding cell size introduces a fundamental trade-off between watermark capacity, embedding granularity, and extraction reliability. Nevertheless, the parameter configurations employed throughout the experimental evaluation satisfy the geometric requirements of the proposed embedding mechanism and therefore do not impose practical limitations on the considered image, audio, and video watermarking frameworks.

### Minimum Capacity Requirements for Cryptographic Verification

A limitation of the proposed integrity-preservation models based on cryptographic hash functions arises from the size of the generated verification information. More specifically, the deployment of SHA-256 and HMAC-SHA256 produces a fixed-length digest of 256 bits, which must be embedded into the digital object. Consequently, the host region selected for embedding must provide sufficient capacity to accommodate the complete verification information.

Let  $L_H$  denote the length of the cryptographic digest. For the employed SHA-256 and HMAC-SHA256 mechanisms,

$$L_H = 256 \text{ bits}. \quad (8.8)$$

Therefore, the corresponding host block must contain at least

$$C_{\min} \geq 256 \quad (8.9)$$

available embedding positions in order to store the complete verification code. As a result, excessively small image blocks, audio blocks, or other embedding regions may

not provide sufficient capacity for the direct storage of the generated authentication information.

This limitation introduces a trade-off between localization granularity and embedding capacity. Smaller blocks improve localization accuracy by enabling finer identification of manipulated regions, whereas larger blocks provide sufficient space for embedding cryptographic verification information. Nevertheless, the image and audio block configurations utilized throughout the experimental evaluation satisfy the minimum capacity requirements of the employed cryptographic primitives and therefore do not impose practical limitations on the proposed models.

### **Video Processing Constraints**

The proposed video authenticity-preservation framework inherits some limitations from its cryptographic verification mechanism. More specifically, the global hash-chain representation embedded into the audio stream requires sufficient embedding capacity within the host audio signal. Since the employed SHA-256 mechanism generates a 256-bit digest, at least 256 audio samples must be available for the storage of the integrity information. Consequently, extremely short audio segments or video sequences with limited audio content may restrict the direct deployment of the proposed verification mechanism without additional compression or fragmentation of the generated hash values.

Furthermore, although the proposed framework exhibits favorable scalability characteristics, its computational requirements increase with video resolution and duration. The generation of cryptographic hashes for all video frames and the subsequent embedding and extraction procedures introduce additional processing overhead, particularly for high-resolution video content and long video sequences.

For the proposed cross-frame permutation-based integrity-preservation framework, an additional challenge arises from the generation and management of large permutation structures. As the number of frames and verification blocks increases, the size of the corresponding permutation mappings also grows, leading to additional memory and computational requirements during both embedding and verification phases. Moreover, temporal attack analysis introduces further computational overhead. While spatial tampering can often be localized by examining only the affected verification relationships, the detection of frame insertion, frame deletion, and frame duplication attacks may require the verification of all blocks belonging to the manip-

ulated frames in order to accurately identify the temporal inconsistencies introduced by the attack.

Although these limitations do not affect the correctness or reliability of the proposed methodologies, they may increase processing times for large-scale video collections and ultra-high-resolution content. Consequently, further optimization of the permutation generation procedures and parallelization of the verification process constitute promising directions for future work.

### **PDF Watermarking Constraints**

The proposed PDF authenticity-preservation framework presents several limitations that originate from the characteristics of the PDF document format itself. More specifically, the embedding process relies on the modification of text-color information and therefore requires the existence of sufficient text objects within the protected document. Consequently, documents consisting primarily of images, scanned pages, or graphical content may provide limited embedding capacity compared to text-rich documents.

Furthermore, the proposed methodology assumes access to the structural information of the PDF document. Operations that preserve the original document structure, such as metadata modifications, digital signatures, page rotation, or scaling operations, do not affect watermark extraction. However, complete document re-rendering procedures, including PDF-to-image conversion followed by document reconstruction, remove the original text objects and consequently destroy the embedded watermark information. Although this behavior is desirable from an integrity-preservation perspective, it limits the ability of the framework to preserve authenticity information after severe document format transformations.

An additional challenge originates from the embedding capacity requirements of the generated QR-code watermark. As the amount of authentication information increases, larger QR-code structures may be required, increasing the number of text modifications that must be performed within the document. Therefore, a trade-off exists between watermark capacity and the availability of suitable embedding locations.

Finally, the computational requirements of the framework increase with the number of document pages, since watermark generation, embedding, extraction, and verification are performed independently for each page. Although the experimental evalu-

ation demonstrated practical execution times for the considered document sizes, very large document collections may require additional optimization and parallelization strategies to improve processing efficiency.

Nevertheless, these limitations do not affect the correctness of the proposed methodology and primarily arise from the structural properties of the PDF format rather than from the watermarking mechanism itself.



# CHAPTER 9

## CONCLUDING REMARKS

---

### 9.1 Research Results

### 9.2 Lessons Learned

### 9.3 Future Research

---

### 9.1 Research Results

The research conducted in this dissertation demonstrated that information hiding techniques can effectively be utilized for both authenticity and integrity preservation across multiple digital object categories, including images, audio signals, videos, and PDF documents. The obtained results verified that the proposed methodologies successfully achieve the primary objectives of robustness, imperceptibility, authenticity verification, tamper detection, and tamper localization.

For authenticity preservation, the proposed watermarking frameworks achieved high robustness against a wide range of signal-processing, geometric, temporal, and document-processing attacks while maintaining satisfactory visual and perceptual quality. The utilization of Self-Inverting Permutations, code-based watermark representations, cryptographic hash-chain mechanisms, and QR-code watermark structures enabled reliable ownership verification and authenticity attestation under adverse operating conditions.

For integrity preservation, the proposed deterministic verification frameworks demonstrated highly accurate tamper detection and localization capabilities. The de-

ployment of cryptographic hash functions, cross-block verification relationships, cross-frame permutations, and audio-assisted verification mechanisms enabled the reliable identification of malicious modifications in images, audio signals, and videos. Unlike learning-based approaches, the proposed methodologies do not require training data and provide explainable verification decisions based on deterministic cryptographic operations.

The experimental evaluation further confirmed the practical applicability of the proposed models. High imperceptibility was achieved across all digital object categories, while the obtained robustness and integrity-preservation results demonstrated the effectiveness of the proposed architectures against realistic attack scenarios. Furthermore, the computational analysis indicated that the proposed methodologies exhibit favorable scalability characteristics and remain suitable for practical deployment.

Overall, the research results validate the effectiveness of the proposed authenticity and integrity preservation frameworks and demonstrate that the combination of combinatorial watermark representations, cryptographic verification mechanisms, and cross-object authentication relationships constitutes a reliable approach for protecting modern digital objects against unauthorized modifications and authenticity-related threats.

## **9.2 Lessons Learned**

The research conducted throughout this dissertation provided several important insights regarding the design of authenticity and integrity preservation systems for digital objects. One of the primary lessons learned is that no single watermarking strategy can simultaneously maximize robustness, imperceptibility, payload capacity, and fragility. Instead, the design of a watermarking framework inevitably involves trade-offs that must be carefully balanced according to the intended application and security requirements.

An important lesson learned concerns the importance of watermark representation. The conducted research demonstrated that representing watermark information as structured codes composed of Self-Inverting Permutations provides greater flexibility, capacity, and uniqueness compared to traditional watermark representations. Furthermore, the obtained results revealed that richer watermark representations can

strengthen authenticity verification capabilities while maintaining satisfactory imperceptibility and robustness characteristics.

Another lesson learned is the importance of integrating cryptographic mechanisms with watermarking techniques. The combination of watermarking and cryptographic verification proved particularly effective for integrity preservation, as cryptographic hashes provide deterministic verification capabilities while watermarking enables the distribution of authentication information within the protected digital object. The obtained results demonstrated that such hybrid approaches can offer reliable tamper detection and localization while maintaining explainability and avoiding the uncertainty often associated with learning-based solutions.

The conducted experiments also highlighted the significance of exploiting structural relationships within digital objects. Cross-block, cross-frame, and cross-modal verification relationships were shown to substantially strengthen authenticity and integrity guarantees by creating dependencies among different object components. These relationships proved particularly effective for detecting localized manipulations, temporal attacks, and coordinated modifications affecting multiple parts of a digital object.

Finally, the research confirmed that practical applicability is as important as robustness and security. High detection accuracy or watermark robustness alone is insufficient if a framework cannot be efficiently deployed in real-world environments. Consequently, scalability, computational efficiency, and implementation complexity must be considered as fundamental design objectives alongside security and reliability requirements.

### **9.3 Future Research**

Although the proposed frameworks demonstrated satisfactory performance for authenticity and integrity preservation, several research challenges remain open. One important direction concerns the further evolution of the proposed code-based watermark representation model. Throughout this dissertation, watermark information was represented through collections of Self-Inverting Permutations encoding integer sequences. Future research may investigate alternative structured watermark representations capable of supporting larger payloads, stronger uniqueness properties, and

richer authentication information while preserving robustness and imperceptibility.

The emergence of generative artificial intelligence introduces additional research challenges. Modern content-generation systems are capable of producing highly realistic synthetic images, audio recordings, videos, and documents that may be indistinguishable from authentic content. Future research should therefore investigate the integration of the proposed watermarking and verification frameworks with AI-generated content attribution systems in order to support provenance verification, synthetic-content identification, and multimedia forensic analysis.

The proposed video integrity-preservation framework also opens new research opportunities regarding temporal verification mechanisms. Future work may focus on the development of more efficient permutation generation procedures, adaptive frame-verification strategies, and scalable architectures capable of supporting long-duration videos and large multimedia repositories. Similarly, optimization techniques and parallel processing strategies could be investigated to further improve the suitability of the proposed methodologies for real-time applications and resource-constrained environments.

Finally, the obtained results indicate that the proposed methodologies possess significant potential for deployment in digital forensic applications. Future research may therefore investigate their integration into forensic analysis platforms, evidence management systems, and trusted digital archiving infrastructures. The combination of watermarking, cryptographic verification, and multimedia provenance mechanisms may contribute towards the development of comprehensive frameworks capable of supporting the authenticity, integrity, and traceability requirements of modern digital ecosystems.

## BIBLIOGRAPHY

---

- [1] M. Chroni, S. D. Nikolopoulos, and L. Palios, “Encoding watermark numbers as reducible permutation graphs using self-inverting permutations,” *Discrete Applied Mathematics*, vol. 250, pp. 145–164, 2018.
- [2] W. Shafik, “Generative adversarial networks: security, privacy, and ethical considerations,” in *Generative artificial intelligence (AI) approaches for industrial applications*. Springer, 2025, pp. 93–117.
- [3] D. A. M. Gosai, “International journal of computer applications,” *International Journal of Computer Applications*, vol. 975, p. 8887, 2015.
- [4] R. Caldelli, F. Filippini, and R. Becarelli, “Reversible watermarking techniques: An overview and a classification,” *EURASIP Journal on Information Security*, vol. 2010, no. 1, p. 134546, 2010.
- [5] G. Nehru and P. Dhar, “A detailed look of audio steganography techniques using lsb and genetic algorithm approach,” *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 1, p. 402, 2012.
- [6] L. Yang, H. Wang, Y. Zhang, M. He, and J. Li, “An adaptive video watermarking robust to social platform transcoding and hybrid attacks,” *Signal Processing*, vol. 224, p. 109588, 2024.
- [7] R. Singh, R. Pal, and D. Joshi, “Optimal frame selection-based watermarking using a meta-heuristic algorithm for securing video content,” *Computers and Electrical Engineering*, vol. 121, p. 109857, 2025.
- [8] H. Sun, Z. Zhang, Z. Li, and J. Liu, “Large-capacity and robust video watermarking via dwt coefficient separation/reconstruction and multi-scale spatiotemporal fusion,” *Neurocomputing*, p. 131275, 2025.

- [9] S. Roy and A. K. Pal, “An svd based location specific robust color image watermarking scheme using rdwt and arnold scrambling,” *Wireless Personal Communications*, vol. 98, no. 2, pp. 2223–2250, 2018.
- [10] S. S. Jamal, T. Shah, S. Farwa, and M. U. Khan, “A new technique of frequency domain watermarking based on a local ring,” *Wireless Networks*, vol. 25, no. 4, pp. 1491–1503, 2019.
- [11] Q. Ying, J. Lin, Z. Qian, H. Xu, and X. Zhang, “Robust digital watermarking for color images in combined dft and dt-cwt domains,” *Mathematical Biosciences and Engineering*, vol. 16, no. 5, pp. 4788–4801, 2019.
- [12] M. Begum, S. B. Shorif, M. S. S. Uddin, J. Ferdush, T. Jan, A. Barros, and M. Whaiduzzaman, “Image watermarking using discrete wavelet transform and singular value decomposition for enhanced imperceptibility and robustness,” *Algorithms*, vol. 17, no. 1, p. 32, 2024.
- [13] X.-y. Wang, C.-p. Wang, H.-y. Yang, and P.-p. Niu, “A robust blind color image watermarking in quaternion fourier transform domain,” *Journal of Systems and Software*, vol. 86, no. 2, pp. 255–277, 2013.
- [14] M. Chroni, S. D. Nikolopoulos, I. Polenakis, and V. Vouronikos, “A code-based multiple watermarking scheme for the preservation of authenticity of forensic images,” in *2023 14th international conference on information, intelligence, systems & applications (IISA)*. IEEE, 2023, pp. 1–8.
- [15] M. Chetan, P. P. Bhat, V. Shet, S. B. Husenbhai, and A. Bhat, “Audio watermarking using modified least significant bit technique,” in *2021 International Conference on Circuits, Controls and Communications (CCUBE)*. IEEE, 2021, pp. 1–5.
- [16] M. Yamni, H. Karmouni, M. Sayyouri, and H. Qjidaa, “Efficient watermarking algorithm for digital audio/speech signal,” *Digital Signal Processing*, vol. 120, p. 103251, 2022.
- [17] N. Cardamone and F. d’Amore, “Dwt and qr code based watermarking for document drm,” in *Digital Forensics and Watermarking: 17th International Workshop, IWDW 2018, Jeju Island, Korea, October 22-24, 2018, Proceedings 17*. Springer, 2019, pp. 137–150.

- [18] Q. B. Dang, K. Louisa, M. Coustaty, M. M. Luqman, and J.-M. Ogier, "A blind document image watermarking approach based on discrete wavelet transform and qr code embedding," in *2019 International conference on document analysis and recognition workshops (ICDARW)*, vol. 8. IEEE, 2019, pp. 1–6.
- [19] Z. Mohsin Arkah, L. Alzubaidi, A. A. Ali, and A. T. Abdulameer, "Digital color documents authentication using qr code based on digital watermarking," in *Intelligent Systems Design and Applications: 18th International Conference on Intelligent Systems Design and Applications (ISDA 2018) held in Vellore, India, December 6-8, 2018, Volume 1*. Springer, 2020, pp. 1093–1101.
- [20] A. F. Qasim, R. Aspin, F. Meziane, and P. Hogg, "Roi-based reversible watermarking scheme for ensuring the integrity and authenticity of dicom mr images," *Multimedia Tools and Applications*, vol. 78, no. 12, pp. 16 433–16 463, 2019.
- [21] M. T. Mathew and P. Geetharanjin, "A reversible watermarking and encryption for ensuring the authenticity, integrity and protection of medical images," in *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*. IEEE, 2020, pp. 340–345.
- [22] D. Laishram and K. Manglem Singh, "A watermarking scheme for source authentication, ownership identification, tamper detection and restoration for color medical images," *Multimedia Tools and Applications*, vol. 80, no. 16, pp. 23 815–23 875, 2021.
- [23] S. Gull, N. A. Loan, S. A. Parah, J. A. Sheikh, and G. M. Bhat, "An efficient watermarking technique for tamper detection and localization of medical images," *Journal of ambient intelligence and humanized computing*, vol. 11, pp. 1799–1808, 2020.
- [24] X. Liu, J. Lou, H. Fang, Y. Chen, P. Ouyang, Y. Wang, B. Zou, and L. Wang, "A novel robust reversible watermarking scheme for protecting authenticity and integrity of medical images," *Ieee Access*, vol. 7, pp. 76 580–76 598, 2019.
- [25] M. M. Sayah, K. M. Redouane, and K. Amine, "Secure transmission and integrity verification for color medical images in telemedicine applications," *Multimedia Tools and Applications*, vol. 81, no. 30, pp. 43 613–43 638, 2022.

- [26] R. Taj, S. Kanwal, A. Alsirhani, F. Alserhani, R. Alanazi, and A. Ali, "Lightweight dual-watermarking framework for medical image authentication and integrity preservation," *Scientific Reports*, 2025.
- [27] R. Bouarroudj, F. Souami, F. Z. Bellala, N. Zerrouki, F. Harrou, and Y. Sun, "Secure and reversible fragile watermarking for accurate authentication and tamper localization in medical images," *Computers and Electrical Engineering*, vol. 123, p. 110072, 2025.
- [28] Q. Yao, K. Xu, T. Li, Y. Zhou, and M. Wang, "A secure image evidence management framework using multi-bits watermark and blockchain in iot environments," *Wireless Networks*, vol. 30, no. 6, pp. 5157–5169, 2024.
- [29] S. S. Sharma and V. Chandrasekaran, "A robust hybrid digital watermarking technique against a powerful cnn-based adversarial attack," *Multimedia Tools and Applications*, vol. 79, no. 43, pp. 32 769–32 790, 2020.
- [30] M. Chroni, A. Fylakis, and S. D. Nikolopoulos, "Watermarking images in the frequency domain by exploiting self-inverting permutations," in *International Conference on Web Information Systems and Technologies*, vol. 2. SCITEPRESS, 2013, pp. 45–54.
- [31] M. Cedillo-Hernandez, A. Cedillo-Hernandez, and F. J. Garcia-Ugalde, "Improving dft-based image watermarking using particle swarm optimization algorithm," *Mathematics*, vol. 9, no. 15, p. 1795, 2021.
- [32] A. Aksay, A. Temizel, and A. E. Cetin, "Camera tamper detection using wavelet analysis for video surveillance," in *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*. IEEE, 2007, pp. 558–562.
- [33] M. A. Kabir, "An efficient low bit rate image watermarking and tamper detection for image authentication," *SN Applied Sciences*, vol. 3, no. 4, p. 400, 2021.
- [34] S. A Hasso and T. Basheer Taha, "A new tamper detection algorithm for video," *Journal of Engineering Science and Technology (JESTEC)*, vol. 15, no. 5, pp. 3375–3387, 2020.



- [35] A. Saglam and A. Temizel, “Real-time adaptive camera tamper detection for video surveillance,” in *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*. IEEE, 2009, pp. 430–435.
- [36] F. Arab, S. M. Abdullah, S. Z. M. Hashim, A. A. Manaf, and M. Zamani, “A robust video watermarking technique for the tamper detection of surveillance systems,” *Multimedia Tools and Applications*, vol. 75, pp. 10 855–10 885, 2016.
- [37] X. Luo, R. Zhan, H. Chang, F. Yang, and P. Milanfar, “Distortion agnostic deep watermarking,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 13 548–13 557.
- [38] A. Chacko and S. Chacko, “Deep learning-based robust medical image watermarking exploiting dct and harris hawks optimization,” *International Journal of intelligent systems*, vol. 37, no. 8, pp. 4810–4844, 2022.
- [39] J. Hayes and G. Danezis, “Generating steganographic images via adversarial training,” *Advances in neural information processing systems*, vol. 30, 2017.
- [40] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, “Hidden: Hiding data with deep networks,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 657–672.
- [41] P. Fernandez, A. Sablayrolles, T. Furon, H. Jégou, and M. Douze, “Watermarking images in self-supervised latent spaces,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 3054–3058.
- [42] R. Sinhal and I. A. Ansari, “Machine learning based multipurpose medical image watermarking,” *Neural Computing and Applications*, vol. 35, no. 31, pp. 23 041–23 062, 2023.
- [43] P. Asmitha, C. Rupa, and S. Nikitha, “Privacy preservation of medical records using ai based watermarking,” in *2024 Second International Conference on Data Science and Information System (ICDSIS)*. IEEE, 2024, pp. 1–6.
- [44] R. Sood, J. Rani, A. Anand, and J. Bedi, “Deep learning-based dual watermarking solution for securing medical images in e-healthcare,” *Knowledge-Based Systems*, p. 114750, 2025.

- [45] S. Zhou, M. Ye, W. Luo, X. Liao, and K. Wei, "Color image steganography using generative adversarial networks with a phased training strategy," in *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, 2025, pp. 142–152.
- [46] C. Zhang, Z. Li, Z. Xu, X. Ren, and L. Zhu, "Sepvemark: Deep separable visual-audio fusion watermarking for source tracing and deepfake detection," in *Proceedings of the 33rd ACM International Conference on Multimedia*, 2025, pp. 8910–8919.
- [47] J. Varghese, O. B. Hussain, S. Subash, and A. Razak, "An effective digital image watermarking scheme incorporating dct, dft and svd transformations," *PeerJ Computer Science*, vol. 9, p. e1427, 2023.
- [48] Z. Yuan, Q. Su, D. Liu, and X. Zhang, "A blind image watermarking scheme combining spatial domain and frequency domain," *The visual computer*, vol. 37, no. 7, pp. 1867–1881, 2021.
- [49] C. Kumar, "Hybrid optimization for secure and robust digital image watermarking with dwt, dct and spiht," *Multimedia Tools and Applications*, vol. 83, no. 11, pp. 31 911–31 932, 2024.
- [50] B. Latreche, H. Naimi, and S. Saadi, "A secure and robust color image watermarking method using svd and gat in the multiresolution dchwt domain," *The Journal of Engineering and Exact Sciences*, vol. 9, no. 10, pp. 17 317–01e, 2023.
- [51] J. Liu, J. Huang, Y. Luo, L. Cao, S. Yang, D. Wei, and R. Zhou, "An optimized image watermarking method based on hd and svd in dwt domain," *IEEE Access*, vol. 7, pp. 80 849–80 860, 2019.
- [52] H. Chaudhary, P. Garg, and V. P. Vishwakarma, "Enhanced medical image watermarking using hybrid dwt-hmd-svd and arnold scrambling," *Scientific Reports*, vol. 15, no. 1, p. 9710, 2025.
- [53] D. Awasthi and V. K. Srivastava, "Performance enhancement of svd based dual image watermarking in wavelet domain using pso and jaya optimization and their comparison under hybrid attacks," *Multimedia Tools and Applications*, vol. 82, no. 23, pp. 35 685–35 717, 2023.

- [54] V. Adul and E. Mwangi, “A robust video watermarking approach based on a hybrid svd/dwt technique,” in *2017 IEEE AFRICON*. IEEE, 2017, pp. 309–313.
- [55] J. Ream, Y. Chu, and D. Schatsky, “Upgrading blockchains: Smart contract use cases in industry,” *Retrieved December*, vol. 12, p. 2017, 2016.
- [56] M. Swan, *Blockchain: Blueprint for a new economy*. ” O’Reilly Media, Inc.”, 2015.
- [57] H. Taherdoost, “Smart contracts in blockchain technology: A critical review,” *Information*, vol. 14, no. 2, p. 117, 2023.
- [58] N. Atzei, M. Bartoletti, and T. Cimoli, “A survey of attacks on ethereum smart contracts (sok),” in *Principles of Security and Trust: 6th International Conference, POST 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings 6*. Springer, 2017, pp. 164–186.
- [59] S. Nakamoto and A. Bitcoin, “A peer-to-peer electronic cash system,” *Bitcoin*.—  
URL: <https://bitcoin.org/bitcoin.pdf>, vol. 4, no. 2, p. 15, 2008.
- [60] V. Buterin *et al.*, “A next-generation smart contract and decentralized application platform,” *white paper*, vol. 3, no. 37, pp. 2–1, 2014.
- [61] N. Radziwill, *Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World*. 2016. Dan Tapscott and Alex Tapscott. New York: Penguin Random House. 348 pages. Taylor & Francis, 2018.
- [62] A. M. Antonopoulos, *Mastering bitcoin: unlocking digital crypto-currencies*. O’Reilly, 2022.
- [63] J. Benet, “Ipfsc-content addressed, versioned, p2p file system,” *arXiv preprint arXiv:1407.3561*, 2014.
- [64] D. Debadri, “Google scraped image dataset,” Tech. Rep. [Online]. Available: <https://www.kaggle.com/dattadebadri/image-classification/data>
- [65] J. P. Cohen, P. Morrison, and L. Dao, “Covid-19 image data collection,” *arXiv 2003.11597*, 2020. [Online]. Available: <https://github.com/ieee8023/covid-chestxray-dataset>

- [66] E. Soares, P. Angelov, S. Biaso, M. H. Froes, and D. K. Abe, "Sars-cov-2 ct-scan dataset: A large dataset of real patients ct scans for sars-cov-2 identification," *medRxiv*, 2020.
- [67] M. Nickparvar, "Brain tumor mri dataset," 2026. [Online]. Available: <https://www.kaggle.com/dsv/14832123>
- [68] K. Pogorelov, K. R. Randel, C. Griwodz, S. L. Eskeland, T. de Lange, D. Johansen, C. Spampinato, D.-T. Dang-Nguyen, M. Lux, P. T. Schmidt, M. Riegler, and P. Halvorsen, "Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection," in *Proceedings of the 8th ACM on Multimedia Systems Conference (MMSYS)*, 2017, pp. 164–169.
- [69] N. Li, T. Li, C. Hu, K. Wang, and H. Kang, "A benchmark of ocular disease intelligent recognition: One shot for multi-disease detection," in *International symposium on benchmarking, measuring and optimization*. Springer, 2020, pp. 177–193.
- [70] pexels. (2023) Forensic images. [Online]. Available: <https://www.pexels.com/search/crime%20scene/>
- [71] H. Chen, W. Xie, A. Vedaldi, and A. Zisserman, "Vggsound: A large-scale audio-visual dataset," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 721–725.
- [72] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6479–6488.
- [73] P. Singh, K. J. Devi, T. K. Nizami, C. S. Prakash, H. K. Thakkar, S. A. Hussain, and S. Mallik, "Ensuring integrity and security of medical image transmission in iomt using highly imperceptible and robust watermarking approach," *Scientific Reports*, vol. 15, no. 1, p. 26058, 2025.
- [74] P. Singh, K. J. Devi, H. K. Thakkar, M. Bilal, A. Nayyar, and D. Kwak, "Robust and secure medical image watermarking for edge-enabled e-healthcare," *IEEE Access*, vol. 11, pp. 135 831–135 845, 2023.

- [75] P. Khare and V. K. Srivastava, "A secured and robust medical image watermarking approach for protecting integrity of medical images," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 2, p. e3918, 2021.
- [76] R. Sinhal, S. Sharma, I. A. Ansari, and V. Bajaj, "Multipurpose medical image watermarking for effective security solutions," *Multimedia Tools and Applications*, vol. 81, no. 10, pp. 14 045–14 063, 2022.
- [77] A. U. Rahman, K. Sultan, D. Musleh, N. Aldhaffer, A. Alqahtani, and M. Mahmud, "Robust and fragile medical image watermarking: a joint venture of coding and chaos theories," *Journal of healthcare engineering*, vol. 2018, no. 1, p. 8137436, 2018.
- [78] H. S. Alshanbari, "Medical image watermarking for ownership & tamper detection," *Multimedia tools and applications*, vol. 80, no. 11, pp. 16 549–16 564, 2021.
- [79] A. Rai and H. V. Singh, "Machine learning-based robust watermarking technique for medical image transmitted over lte network," *Journal of Intelligent Systems*, vol. 27, no. 1, pp. 105–114, 2018.
- [80] M. Cedillo-Hernandez, A. Cedillo-Hernandez, M. Nakano-Miyatake, and H. Perez-Meana, "Improving the management of medical imaging by using robust and secure dual watermarking," *Biomedical Signal Processing and Control*, vol. 56, p. 101695, 2020.
- [81] L. P. Meda and E. Rayachoti, "Enhancing telemedicine security in advanced methods for tamper detection and recovery in color medical imaging," *IEEE Access*, vol. 13, pp. 217 542–217 555, 2025.
- [82] S. Saxena, A. Singh, and S. Tiwari, "Prediction model for digital image tampering using customised deep neural network techniques," *International Journal of System Assurance Engineering and Management*, vol. 16, no. 10, pp. 3263–3271, 2025.
- [83] J. Zhang, X. Huang, Y. Liu, Y. Han, and Z. Xiang, "Gan-based medical image small region forgery detection via a two-stage cascade framework," *Plos one*, vol. 19, no. 1, p. e0290303, 2024.

- [84] U. Kumaran, M. Gurupriya, H. R. Thodathara, A. Vijjapu, G. Harish *et al.*, “Enhancing healthcare image record security via cnn-based tamper detection, watermarking, and digital signatures,” in *2024 2nd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS)*. IEEE, 2024, pp. 25–34.
- [85] E. Salah, Z. Narima, M. M. Sayah, K. Amine, K. Redouane, A. K. Sahu, and N. Mimoune, “Blind audio watermarking for medical data authentication using fractional charlier transform and adaptive dithered quantization index modulation,” *Systems and Soft Computing*, p. 200464, 2026.
- [86] S. Priyadharsini and A. Kanhe, “A hybrid chebyshev-svd based approach for robust audio watermarking application,” *Signal Processing*, p. 110279, 2025.
- [87] W. Huan, S. Li, Z. Qian, and X. Zhang, “Exploring stable coefficients on joint sub-bands for robust video watermarking in dt cwt domain,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, pp. 1955–1965, 2021.
- [88] X. Luo, Y. Li, H. Chang, C. Liu, P. Milanfar, and F. Yang, “Dvmark: a deep multiscale framework for video watermarking,” *IEEE Transactions on Image Processing*, vol. 34, pp. 4371–4385, 2023.
- [89] Z. Zhang, H. Wang, G. Wang, and X. Wu, “Hide and track: Towards blind video watermarking network in frequency domain,” *Neurocomputing*, vol. 579, p. 127435, 2024.
- [90] A. Cedillo-Hernandez, L. Velazquez-Garcia, M. Cedillo-Hernandez, I. Dominguez-Jimenez, and D. Conchouso-Gonzalez, “Robust jnd-guided video watermarking via adaptive block selection and temporal redundancy,” *Mathematics*, vol. 13, no. 15, p. 2493, 2025.
- [91] H. Wang, H. Wang, M. He, F. Zhang, and J. Xia, “Robust blind video watermarking against digital editing and camcording,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 35, no. 11, pp. 11 068–11 082, 2025.
- [92] M. He, H. Wang, F. Zhang, S. M. Abdullahi, and L. Yang, “Robust blind video watermarking against geometric deformations and online video sharing platform processing,” *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 6, pp. 4702–4718, 2022.

- [93] Y. Zhang, J. Ni, W. Su, and X. Liao, "A novel deep video watermarking framework with enhanced robustness to h. 264/avc compression," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 8095–8104.
- [94] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 657–672.
- [95] M. Tancik, B. Mildenhall, and R. Ng, "Stegastamp: Invisible hyperlinks in physical photographs," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2117–2126.
- [96] M. Masoumi and S. Amiri, "A blind scene-based watermarking for video copyright protection," *AEU-International Journal of Electronics and Communications*, vol. 67, no. 6, pp. 528–535, 2013.
- [97] W. Chun-peng, W. Xing-yuan, and X. Zhi-qiu, "Geometrically invariant image watermarking based on fast radial harmonic fourier moments," *Signal Processing: Image Communication*, vol. 45, pp. 10–23, 2016.
- [98] B. Ma, L. Chang, C. Wang, J. Li, X. Wang, and Y.-Q. Shi, "Robust image watermarking using invariant accurate polar harmonic fourier moments and chaotic mapping," *Signal processing*, vol. 172, p. 107544, 2020.
- [99] M. Du, T. Luo, H. Xu, Y. Song, C. Wang, and L. Li, "Robust hdr video watermarking method based on the hvs model and t-qr," *Multimedia Tools and Applications*, vol. 81, no. 23, pp. 33 375–33 395, 2022.
- [100] X. Chang, B. Chen, W. Ding, and X. Liao, "A dnn robust video watermarking method in dual-tree complex wavelet transform domain," *Journal of Information Security and Applications*, vol. 85, p. 103868, 2024.
- [101] X. Liu, Q. Dang, H. Wang, X. Deng, X. Fan, C. Yang, Z. Chen, and H. Fang, "An adversarial contrastive learning based cross-modality zero-watermarking scheme for dibr 3d video copyright protection," *Neurocomputing*, vol. 637, p. 130068, 2025.
- [102] A. Cedillo-Hernandez, L. Velazquez-Garcia, F. J. Garcia-Ugalde, and M. Cedillo-Hernandez, "Deep learning-based video watermarking: A robust framework for

- spatial–temporal embedding and retrieval,” *Future Internet*, vol. 18, no. 2, p. 104, 2026.
- [103] M. Aloraini, M. Sharifzadeh, and D. Schonfeld, “Sequential and patch analyses for object removal video forgery detection and localization,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 3, pp. 917–930, 2020.
- [104] Y. Rodriguez-Ortega, D. M. Ballesteros, and D. Renza, “Copy-move forgery detection (cmfd) using deep learning for image and video forensics,” *Journal of imaging*, vol. 7, no. 3, p. 59, 2021.
- [105] M. Raveendra and K. Nagireddy, “Tamper video detection and localization using an adaptive segmentation and deep network technique,” *Journal of Visual Communication and Image Representation*, vol. 82, p. 103401, 2022.
- [106] G. Singh and K. Singh, “Video frame and region duplication forgery detection based on correlation coefficient and coefficient of variation,” *Multimedia Tools and Applications*, vol. 78, no. 9, pp. 11 527–11 562, 2019.
- [107] D.-N. Zhao, R.-K. Wang, and Z.-M. Lu, “Inter-frame passive-blind forgery detection for video shot based on similarity analysis,” *Multimedia Tools and Applications*, vol. 77, pp. 25 389–25 408, 2018.
- [108] Y. Yao, Y. Shi, S. Weng, and B. Guan, “Deep learning for detection of object-based forgery in advanced video,” *Symmetry*, vol. 10, no. 1, p. 3, 2017.
- [109] L. Su, T. Huang, and J. Yang, “A video forgery detection algorithm based on compressive sensing,” *Multimedia Tools and Applications*, vol. 74, pp. 6641–6656, 2015.
- [110] K. Sitara and B. Mehtre, “Detection of inter-frame forgeries in digital videos,” *Forensic science international*, vol. 289, pp. 186–206, 2018.
- [111] J. Bakas, R. Naskar, and R. Dixit, “Detection and localization of inter-frame video forgeries based on inconsistency in correlation distribution between haralick coded frames,” *Multimedia Tools and Applications*, vol. 78, pp. 4905–4935, 2019.
- [112] J. H. Hong, Y. Yang, and B. T. Oh, “Detection of frame deletion in hevc-coded video in the compressed domain,” *Digital Investigation*, vol. 30, pp. 23–31, 2019.



- [113] Y. Abdalla, M. T. Iqbal, and M. Shehata, "Convolutional neural network for copy-move forgery detection," *Symmetry*, vol. 11, no. 10, p. 1280, 2019.
- [114] N. A. S. Al-maweri, W. A. W. Adnan, A. R. Ramli, K. Samsudin, and S. M. S. A. A. Rahman, "Robust digital text watermarking algorithm based on unicode extended characters," *Indian Journal of Science and Technology*, 2016.
- [115] P. Dikanev and Y. Vybornova, "Method for protection of pdf documents against counterfeiting using semi-fragile watermarking," in *2021 International Conference on Information Technology and Nanotechnology (ITNT)*. IEEE, 2021, pp. 1–4.
- [116] S. Ge, Z. Xia, J. Fei, Y. Tong, J. Weng, and M. Li, "A robust document image watermarking scheme using deep neural network," *Multimedia Tools and Applications*, vol. 82, no. 25, pp. 38 589–38 612, 2023.
- [117] B. Singh and M. Sharma, "Watermarking technique for document images using discrete curvelet transform and discrete cosine transform," *Multimedia Tools and Applications*, pp. 1–25, 2024.
- [118] N. A. A. Usop, S. I. Hisham, and J. M. Zain, "An implementing of zigzag pattern in numbering watermarking bits for high detection accuracy of tampers in document," *Indian J. Comput. Sci. Eng*, vol. 13, no. 5, pp. 1735–1749, 2022.
- [119] Y. Zhou, L. Hong, and X. Zhu, "Document image watermarking scheme with adaptive positioning based on discrete wavelet transform and discrete cosine transform," in *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2022, pp. 1–6.

## AUTHOR'S PUBLICATIONS

---

- V. Vouronikos, I. Polenakis, D. Merkouris and S. D. Nikolopoulos, “HELIX: Malware Detection and Classification through Temporal Sequence Modeling of System-Call Groups, *Journal of Computer Virology and Hacking Techniques*, 2026 (accepted).
- I. Polenakis, V. Vouronikos, K. Balis, and S. D. Nikolopoulos, “A code-based multiple watermarking scheme for the preservation of authenticity of digital video,” *Multimedia Tools and Applications*, vol. 85, no. 4, p. 317, 2026.
- I. Polenakis, V. Vouronikos, A. Merolli, and S. D. Nikolopoulos, “A PDF watermarking scheme utilizing invisible QR code embedding in the spatial domain,” *Journal of Computer Security*, vol. 33, no. 6, pp. 363–385, 2025.
- I. Polenakis, V. Vouronikos, N. Vouronikos, M. Chroni, and S. D. Nikolopoulos, “A code-based watermarking scheme for the protection of authenticity of medical images,” in *Optimization, Discrete Mathematics and Applications to Data Sciences*, Springer, 2024, pp. 175–209.
- I. Polenakis, V. Vouronikos, E. Kiafa, M. Chroni, and S. D. Nikolopoulos, “Pro-tain: Protecting the authenticity and integrity properties of NFT images,” in *Proceedings of the International Conference on Computer Systems and Technologies 2024*, pp. 16–23, 2024. – BEST PAPER AWARD
- M. Chroni, S. D. Nikolopoulos, I. Polenakis, and V. Vouronikos, “A code-based multiple watermarking scheme for the preservation of authenticity of forensic images,” in *2023 14th International Conference on Information, Intelligence, Systems & Applications (IISA)*, IEEE, 2023, pp. 1–8.
- M. Chroni, A. Katsi, S. D. Nikolopoulos, I. Polenakis, and V. Vouronikos, “Tamper detection and localization in forensic images,” in *Proceedings of the 24th*

International Conference on Computer Systems and Technologies, pp. 117–122, 2023. – BEST PAPER AWARD

- M. Chroni, S. D. Nikolopoulos, I. Polenakis, and V. Vouronikos, “A multiple watermarking scheme for digital audio signals,” in Proceedings of the 26th Pan-Hellenic Conference on Informatics, pp. 279–284, 2022.
- M. Chroni, S. D. Nikolopoulos, I. Polenakis, and V. Vouronikos, “A repetitive watermarking scheme for digital images based on self-inverting permutations,” in 2022 13th International Conference on Information, Intelligence, Systems & Applications (IISA), IEEE, 2022, pp. 1–8.
- M. Chroni, S. D. Nikolopoulos, I. Polenakis, and V. Vouronikos, “Digital image watermarking using self-inverting permutations—a comparative study,” in 2022 13th International Conference on Information, Intelligence, Systems & Applications (IISA), IEEE, 2022, pp. 1–8.

## SHORT BIOGRAPHY

---

Vasileios Vouronikos was born in Larissa, Greece, in 1997. He received his Diploma in Computer Science and Engineering (Integrated Master) from the Department of Computer Science and Engineering of the University of Ioannina in 2022, with a thesis on algorithmic techniques for Digital Image Watermarking. In the same year, he commenced his doctoral studies in the same department, pursuing research in the areas of watermarking, information hiding, and tamper detection and localization in multimedia content. During his doctoral studies, he participated as a researcher and software engineer in the project “Dioni: Computing Infrastructure for Big-Data Processing and Analysis”, co-financed by the European Union and National Resources under the NSRF 2014–2020 programme. His research has been published in international conferences and Journals and has been recognised with a Best Paper Award at the 23rd and 24th International Conference on Computer Systems and Technologies (CompSysTech’23 and CompSysTech’24). He is currently working as a Software Engineer at Deloitte.