Algorithmic Techniques for Video Watermarking

Student : Konstantinos Balis (Student ID : 4417)
Supervisor : Stavros D. Nikolopoulos

Department of Computer Science & Engineering
University of Ioannina, Ioannina, Greece

Graduation Thesis Presentation
12 July 2023, University of Ioannina, Greece

Structure & Contents

Structure & Contents

4. Evaluation
   - Dataset
   - Attack Vector
   - Experimental Design
   - Measuring Fidelity Imperceptibility
   - Video Authenticity Verification Results
5. Conclusion
   - Potentials and Limitations
   - Future Research

1. Introduction

Video watermarking

Video watermarking is the process of embedding unique identifiers in the components of a video (frames and/or audio) to safeguard intellectual property, authenticate content, monitor and prevent unauthorized tampering.

Types of watermarks :

- Embedding information : logos, text, images, data, patters, QR codes
- Visibility : Visible/Semi-visible or Invisible

Selection of the type of watermark depends on the requirements we have in each case but the goal of watermarking is the same. Developing effective and robust watermarks that can withstand various attacks and enable extraction solely to authorized users to prove ownership of the video.

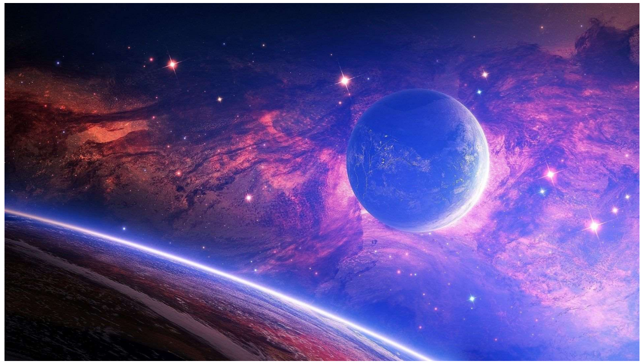
1. Introduction

Authenticity Verification

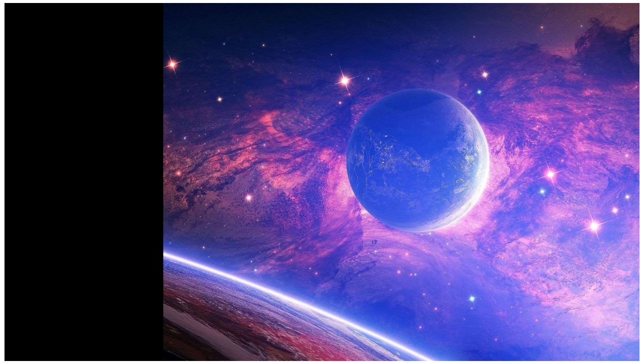
- Authentication is the process of attempting to verify the integrity and authenticity of a watermarked video.

- A digital watermark contains vital information that can be used to establish ownership of the media and validating that the video has not been altered in any way.

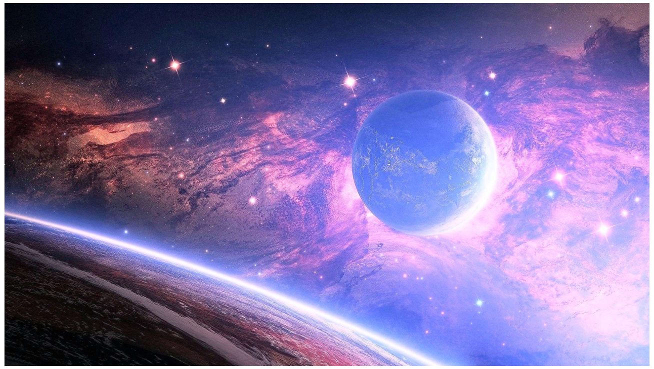
Original Frame



Tampered Frame



Tampered Frame

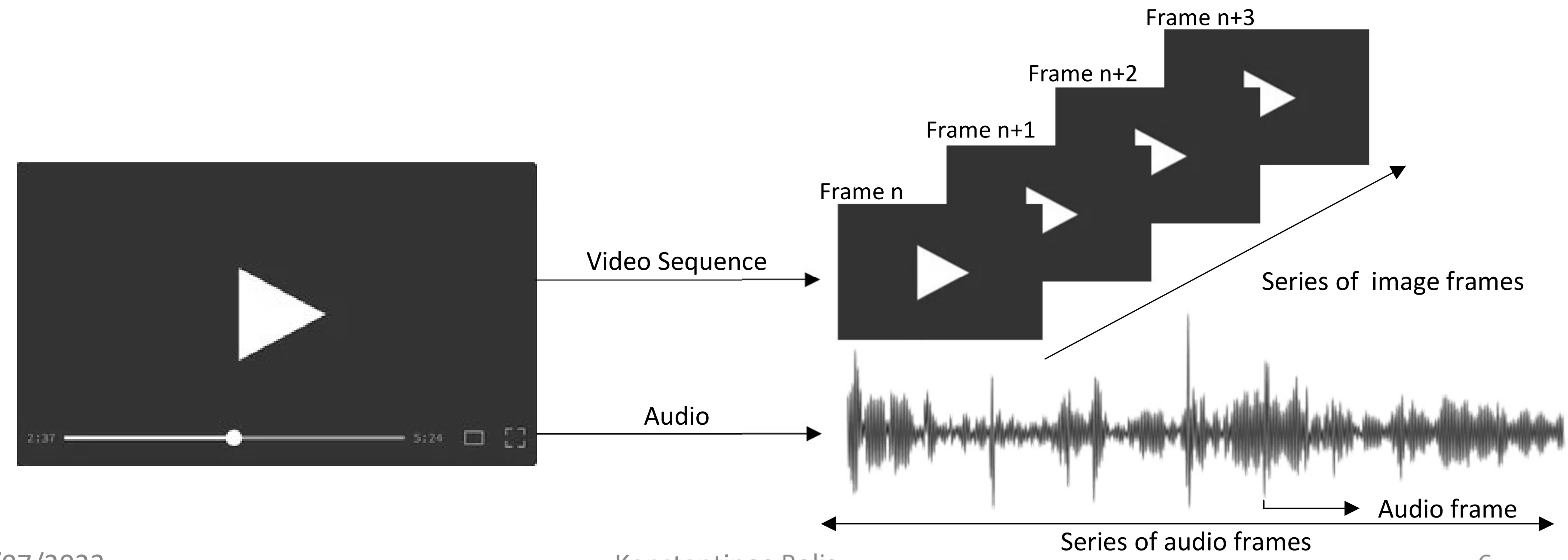


Authenticity verification process should fail to recognize videos that contain tampered frames as authentic

Video Components

The main two components of a video are the video sequence and the audio.

- Video sequence is a series of frames/images that when they are played in a short period of time create the illusion of motion, measured in frames per second (FPS).

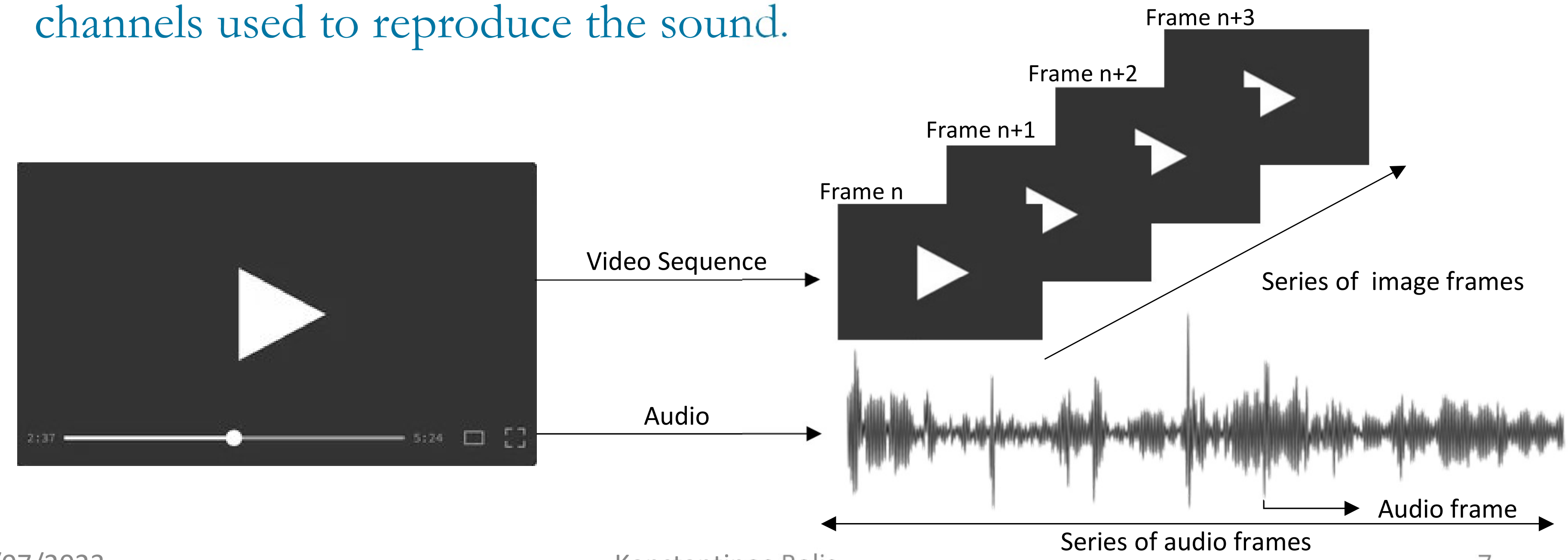


2. Theoretical Background

Video Components

The main two components of a video are the video sequence and the audio.

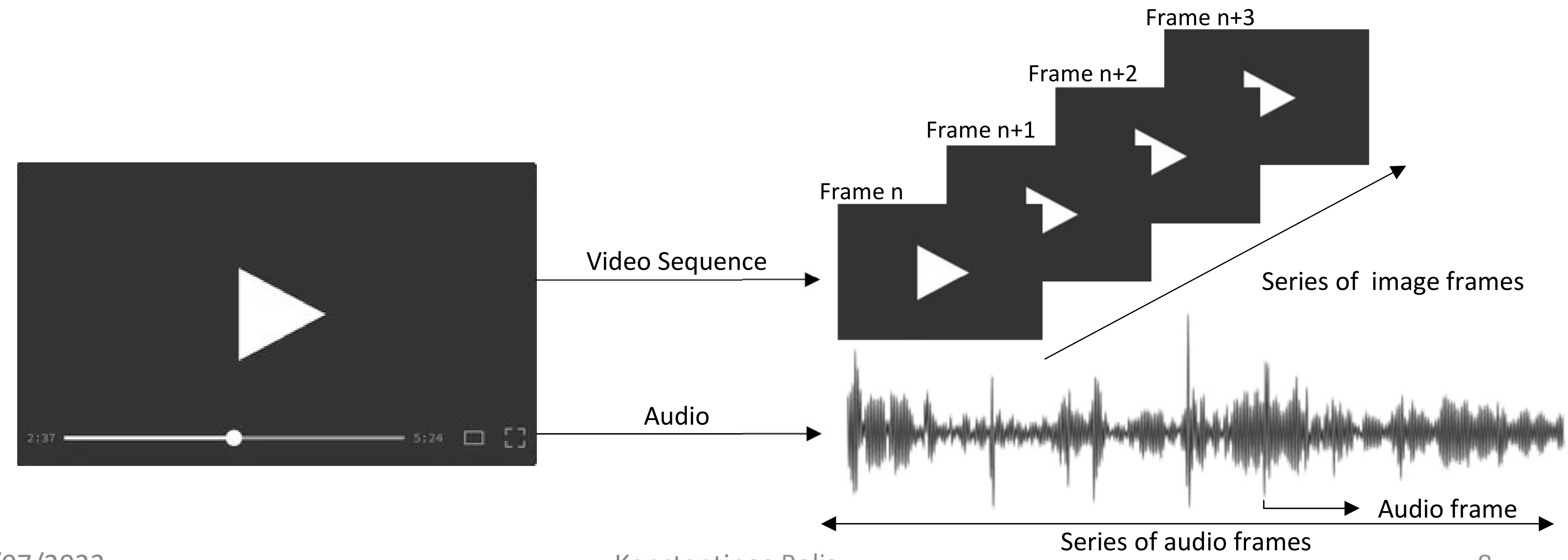
- Audio is the representation of sound synchronized with the frames of the video. Audio can be separated into audio frames that contain information about the sound of the video in certain moments. Each audio frame is consisted of one or more audio samples according to the number of channels used to reproduce the sound.



Video Components

The main two components of a video are the video sequence and the audio.

- Retrieving these components is essential in order to access them and embed watermarks or additional information to the video.



2. Theoretical Background

Encode Integer as Self-Inverting Permutation (SIP)

- Self-Inverting Permutation is the method that will be used to embed watermark into the frames and more specifically the 2D representation of the SIP.

- Permutation is the arrangement of a set of objects in a specific order. In this case the arrangement is done in way that enables both encoding and decoding by following the inverse steps of the algorithm. This allows us to encode an integer during the embedding process and get back that same integer during the extraction process

2. Theoretical Background

The Encoding Process of an Integer as Self-Inverting Permutation

Input: Integer

Compute Binary Representation

Form the Binary Sequence B

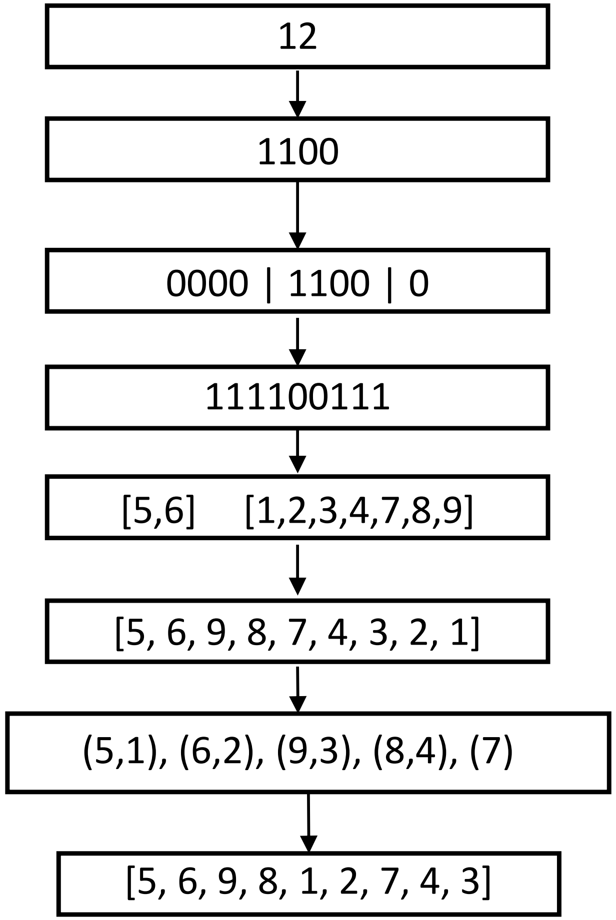
Flip the elements B*

Compute the following sequences:
- X sequence the indices of every 0 in B*
- Y sequence the indices of every 1 in B*

Form Bitonic Sequence $X|Y^R$

Form cycles:
- if size of bitonic sequence even all cycles have length 2
- else all cycles have length 2 except one which has length 1

Construct Self-Inverting Permutation:
- for every index of the permutation insert the other element of the cycle the index belongs



2. Theoretical Background

2D Representation of Self-Inverting Permutation

After encoding the integer as a self-inverting permutation, the next step is to create a representation capable of being embedded into a two-dimensional object as o image frame. The following technique maps the permutation into the cells of a $n \times n$ matrix by marking the cell that is in the same row the index of the permutation suggest and the same column as the element itself.

Continuing with the previous example, on the right is the 2D representation of the Self-Inverting Permutation for integer 12:

index	1	2	3	4	5	6	7	8	9
permutation	5	6	9	8	1	2	7	4	3

				*				
					*			
								*
							*	
*								
	*							
						*		
			*					
		*						

Frequency Domain Watermarking

- Frequency domain watermarking is the process of embedding the watermarking information in the frequency components of the image instead of changing the values of the pixels in the spatial domain.
- Watermarking in the frequency domain offers a couple of advantages in comparison with the spatial domain watermarking. More specifically:
  - ✓ Upon inversion back watermark information has been distributed across the spatial block making it more challenging to locate and extract.
  - ✓ The changes to the pixels that happen in the frequency domain are significantly less observable to the human eye than the changes that happen directly in the pixel in the spatial domain.

Robustness and imperceptibility make frequency domain watermarking the preferred choice for this method.

2. Theoretical Background

Integrity Preservation (Hash)

To ensure integrity preservation of the frames of the video a cryptographic hash function will be employed. Hash values have several desirable qualities to help us identify potential tampering.

- Hash function generate a fixed-sized statistically unique output for a particular set of data.
- Are collision resistant functions. Different inputs, even similar inputs with small modifications, will always result in significantly different hash values.
- Are computationally efficient, regardless of the size of the input.

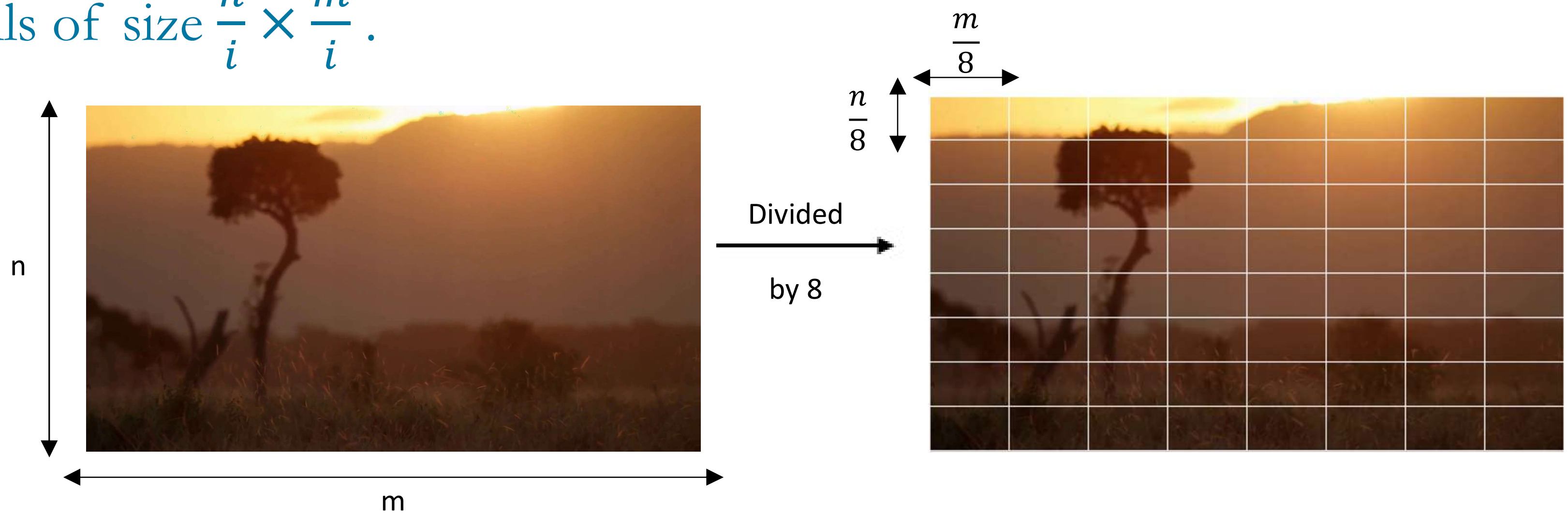
3. The Model

Partitioning Frames into Integrity-Blocks

Partitioning of a frame is the sub-division of a frame into multiple non-overlapping blocks of equal size, called integrity-blocks.

This division of the frame allows us to process each block of the image independently and embed the watermark information.

Dividing a frame $n \times m$ size with an integer i is to create an $i \times i$ grid with cells of size $\frac{n}{i} \times \frac{m}{i}$.



3. The Model

Performing FDW in Image Frames

- The 2D Discrete Fourier Transform (DFT) is used to transform an image from the spatial domain to the frequency domain.

$$F(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y)\, e^{-j2\pi\left(\frac{ux}{N}+\frac{vy}{M}\right)}$$

- Similarly, we can use 2D Inverse DFT to transform the image back to the spatial domain.

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{M-1} F(u,v)\, e^{j2\pi\left(\frac{ux}{N}+\frac{vy}{M}\right)}$$

- Except the transformation of the data, we are interested in the magnitude which represents the amplitude of the corresponding frequency components.

3. The Model

Encoding Multiple Integers as Self-Inverting Permutations

- The choice of watermark in this work is a series of N integers of the same class encoded as 2D representations of their Self-Inverting Permutations, where N is the number of integrity-blocks that will be created. By "same class" we mean integers that their binary representation requires the same number of bits.

  - For example, 4, 5, 6 and 7 all require 3 bits to be represented.

- Following the methodology described about encoding integers we can transform this series of integers into a series of 2D representations of Self-Inverting Permutations to embed to each integrity-block.

$$N \left\{ \begin{pmatrix} 6 \\ 4 \\ 7 \\ \ldots \\ 5 \\ 4 \end{pmatrix} \right. \xrightarrow{\text{SIP}} \begin{pmatrix} [4,5,7,1,2,6,3] \\ [4,7,6,1,5,3,2] \\ [4,5,6,1,2,3,7] \\ \ldots \\ [4,6,7,1,5,2,3] \\ [4,7,6,1,5,3,2] \end{pmatrix} \xrightarrow[\text{SIP}]{\text{2D rep.}} \begin{pmatrix} 2D\_rep(6) \\ 2D\_rep(4) \\ 2D\_rep(7) \\ \ldots \\ 2D\_rep(5) \\ 2D\_rep(4) \end{pmatrix}$$

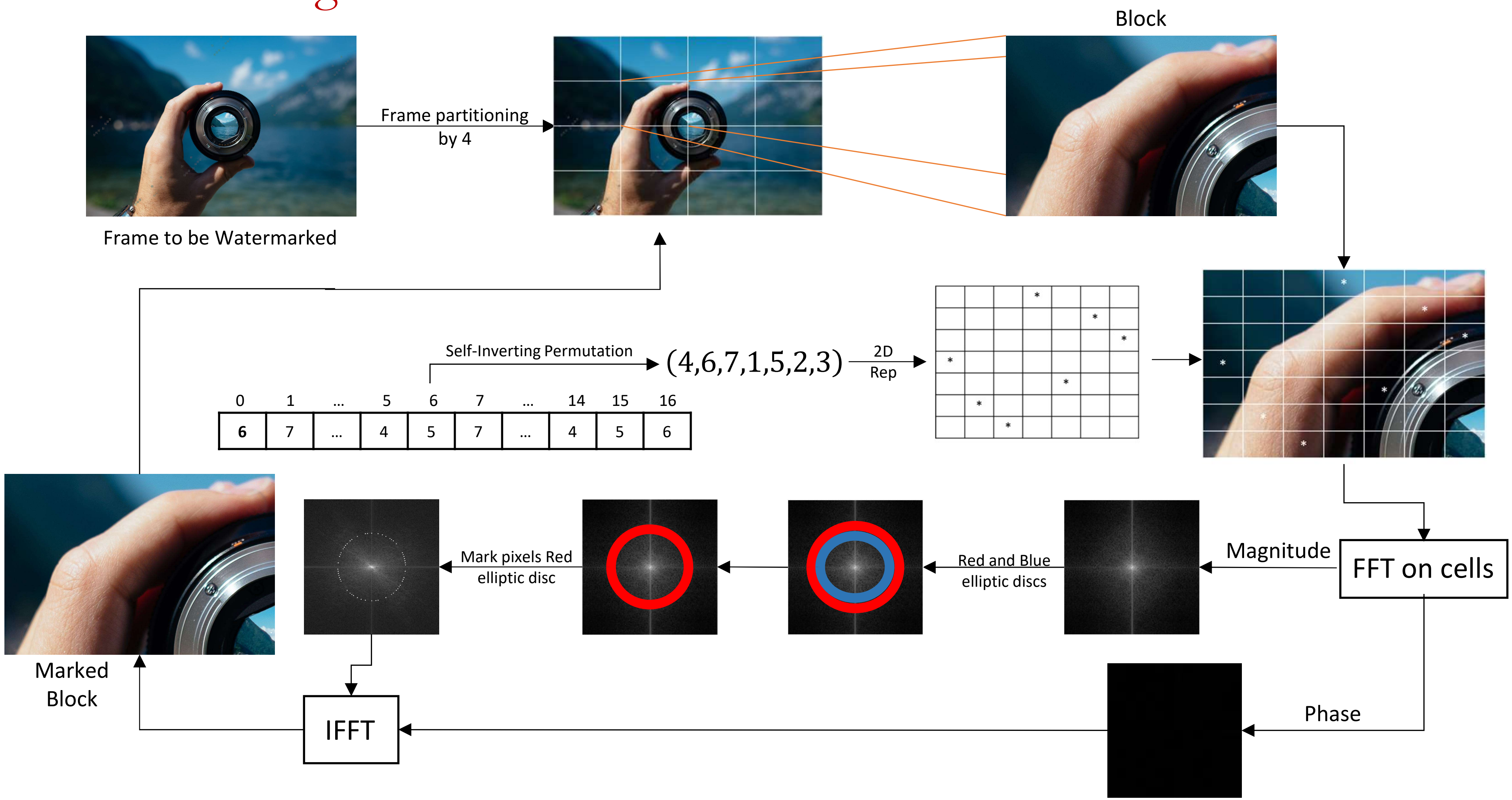
3. The Model

Embedding Codes to Frames

1. Embedding the watermarking information in this method requires the partitioning of the frame into integrity-blocks.
2. For each of the blocks that are created the is a respective integer to be encoded as the 2D representation of the self-inverting permutation of that integer.
3. Then we apply the 2D representation into the block so we can locate the cells we will embed the information.

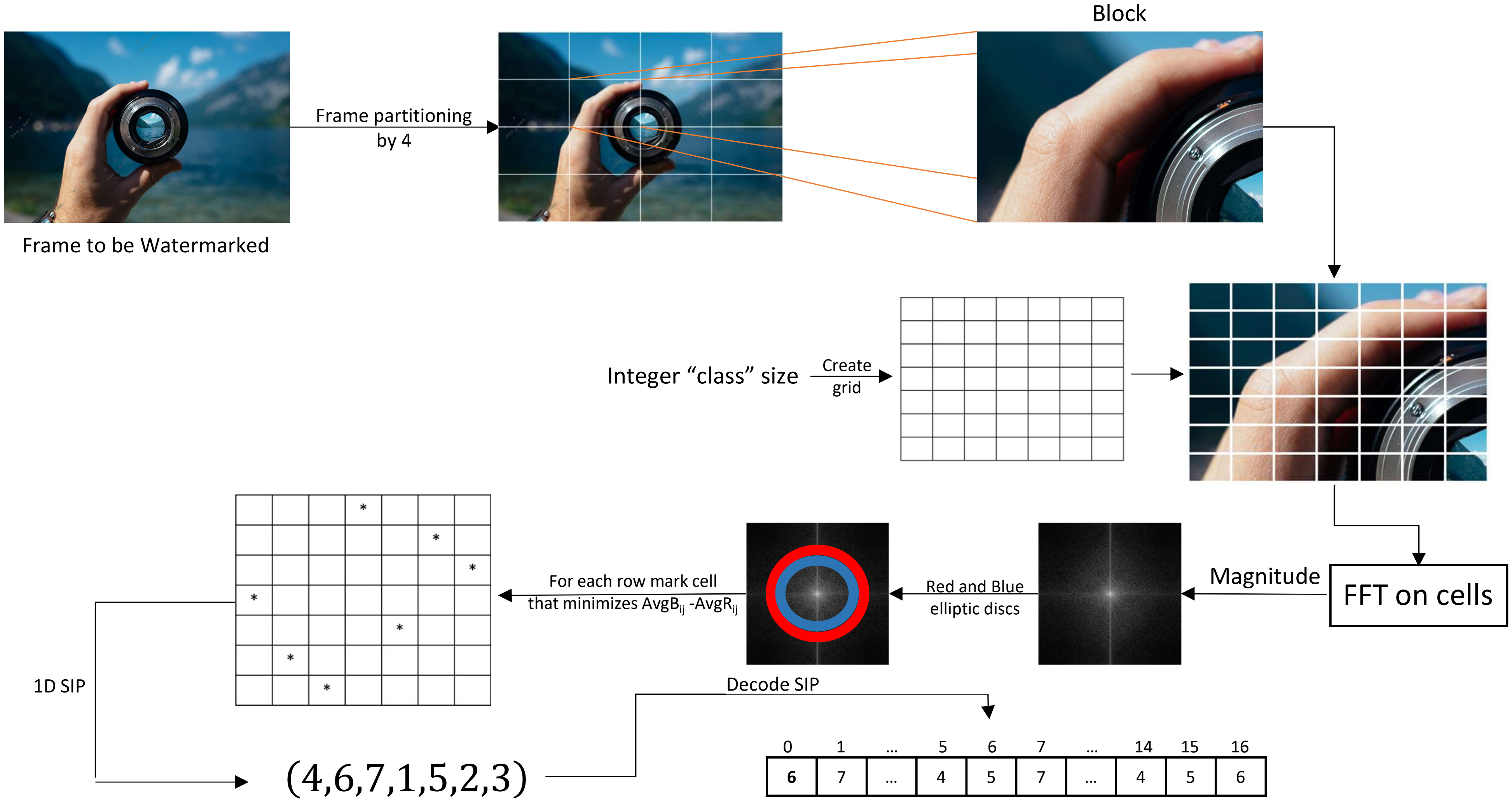
3. The Model

Embedding Codes to Frames



3. The Model

Extract Watermarks from Frames



3. The Model

Preserving Video Integrity

- Hash function and hash values are extremely effective methods to prove integrity of a video or detect modifications in the data video.

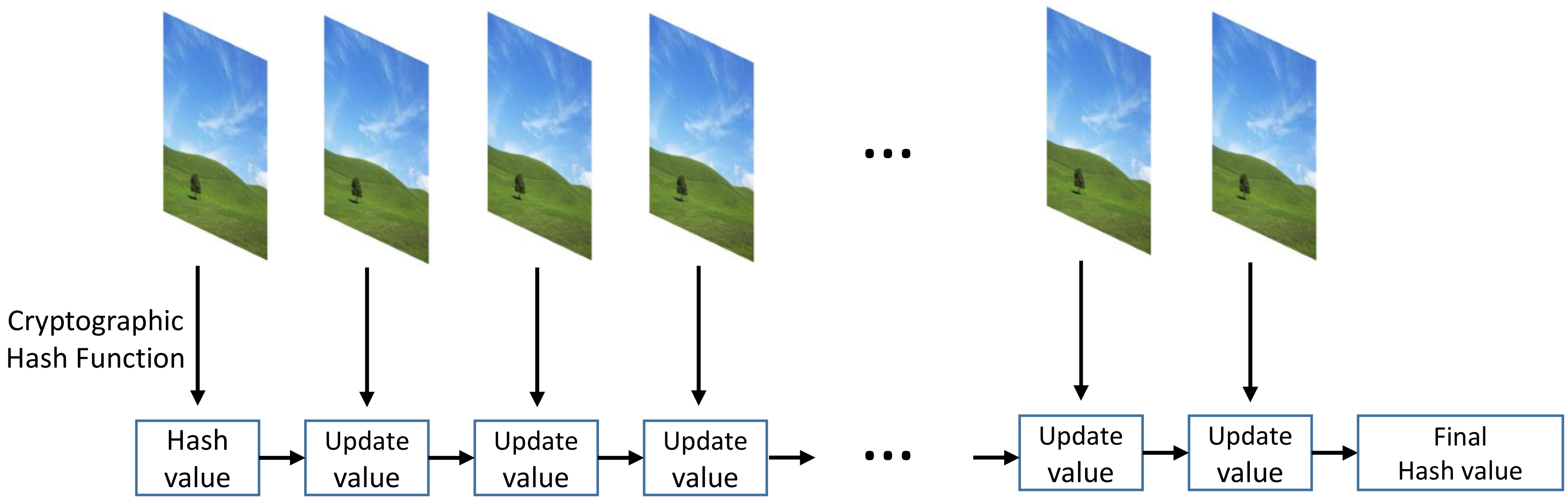
- An algorithm based on hash values will be presented that embeds information about the frames of the watermarked video that upon extraction can verify the authenticity of a video comparing the embedded hash value information about the frames with the hash value of the frames of the video which integrity we try to verify.

3. The Model

Preserving Video Integrity

Hash-Based Video Frame-Chain

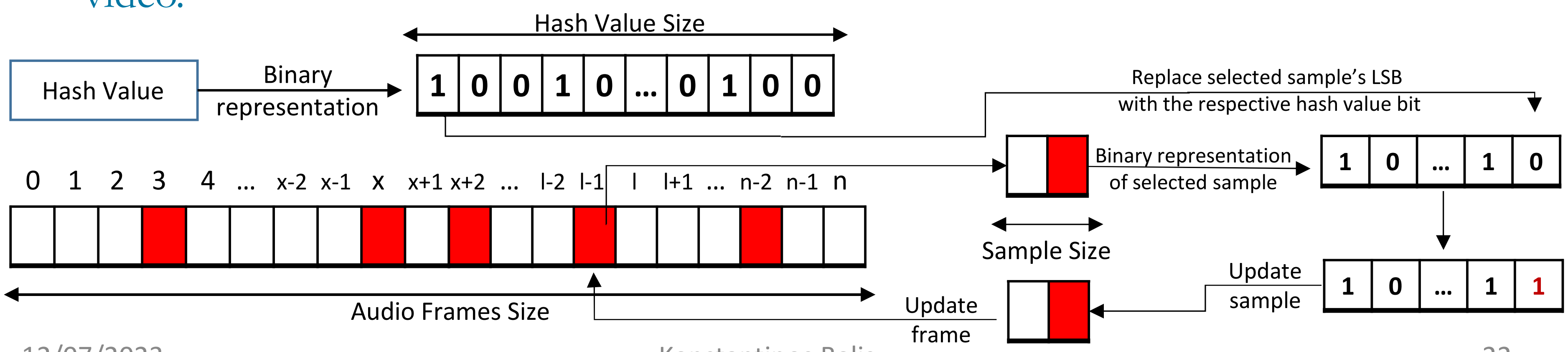
- To encrypt information about the frames with that hash-based technique we will follow the block-chain approach which links the frames together in chronological order to form a chain. That means that every time the hash value is updated it will contain information to all the previous blocks.



Preserving Video Integrity

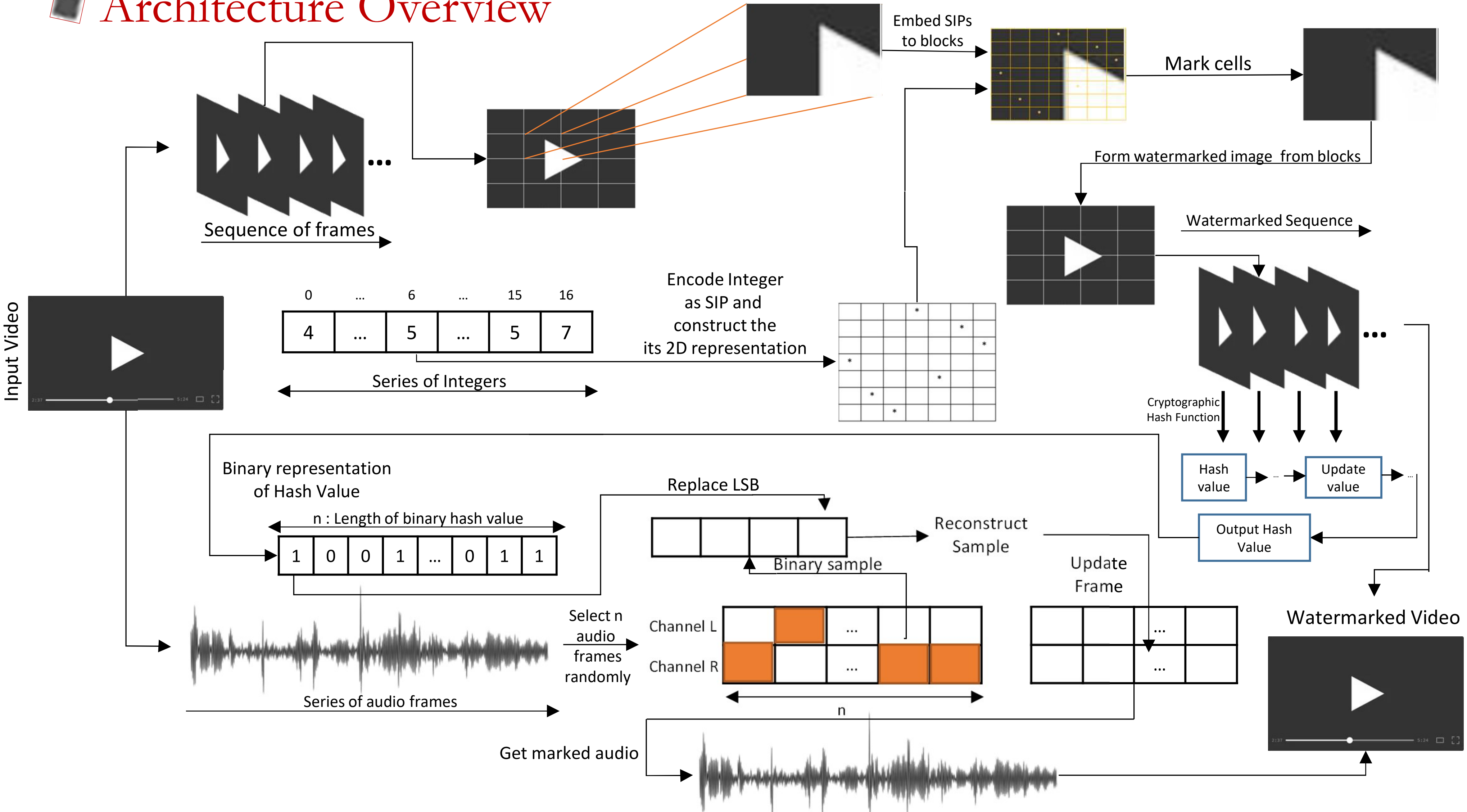
Embedding Integrity Information into the Least Significant Bit (LSB) of Audio Samples

- After computing the hash-value of the frames, calculate its binary representation.
- Select n frames of the audio, where n is the size of the binary representation of the hash, and a sample in each frame to embed a bit of the hash in its least significant bit.
  - Note that the order of the selected frames matter and the user has the necessary information about the correct order of the frames and the marked samples to extract hash value.
- Lastly, update the sample and the frame in the audio to embed the information in the video.



3. The Model

Architecture Overview



4. Evaluation

Dataset

- Dataset: The dataset selected for the evaluation of the method contains 2 videos in mp4 file format of different resolutions. One video “Nature” with 1080p resolution (1980×1080) and “Man at the Sea” with 720p resolution (1280×720).

Nature

Man at the Sea

4. Evaluation

Attack Vector

For the evaluation of the model different attack types used to manipulate frames of the video.

- **Noise Attacks:** These attacks cause random variations in the pixels of the frames of a video. The noise attacks that will be used are the Gaussian noise and Salt & Pepper.
- **Blur Attacks:** These attacks result in the blurring of video frames. Gaussian, Average and Median blur will be used.
- **Enhance Attacks:** These attacks are applying enhancement techniques to the frames of the video. Histogram equalization and Gamma are the enhancement attacks that we will use.
- **Compression Attack:** These attacks compress frames of the video.
- **Crop Attack:** These attacks remove parts of the frame.

4. Evaluation

Experimental Design

- Image Partition and Permutations: The integer used to partition the frames is 8 which creates an 8×8 grid with block size (240×135) for 1080p videos and (160×90) for the 720p videos. The integers used as watermarks were integers that their binary representation has 4 digits [8, 15] and the 2D representation of their Self-Inverting Permutations creates a 9×9 grid.

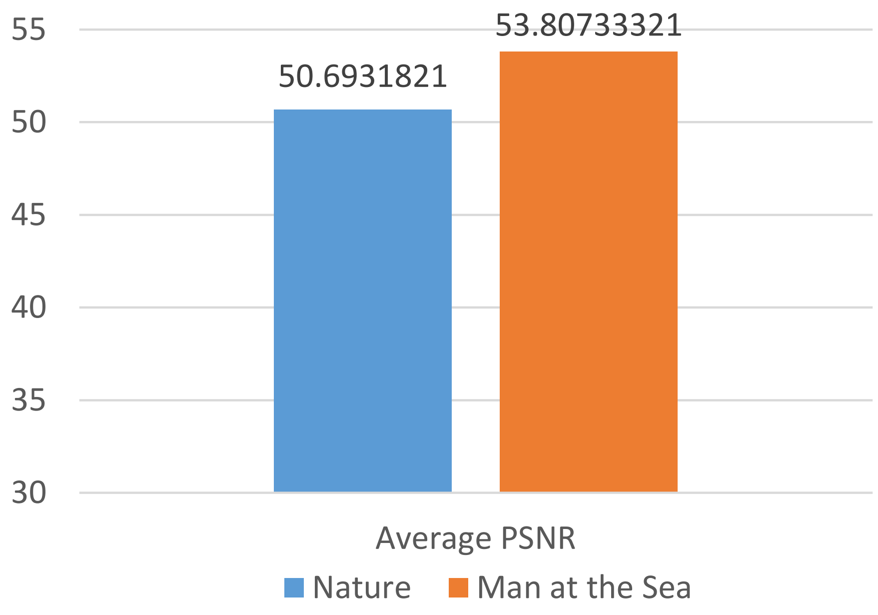
- Hash Function: For the computation of the hash value, the SHA-256 hash algorithm will be used that produces a 256bit hash value.

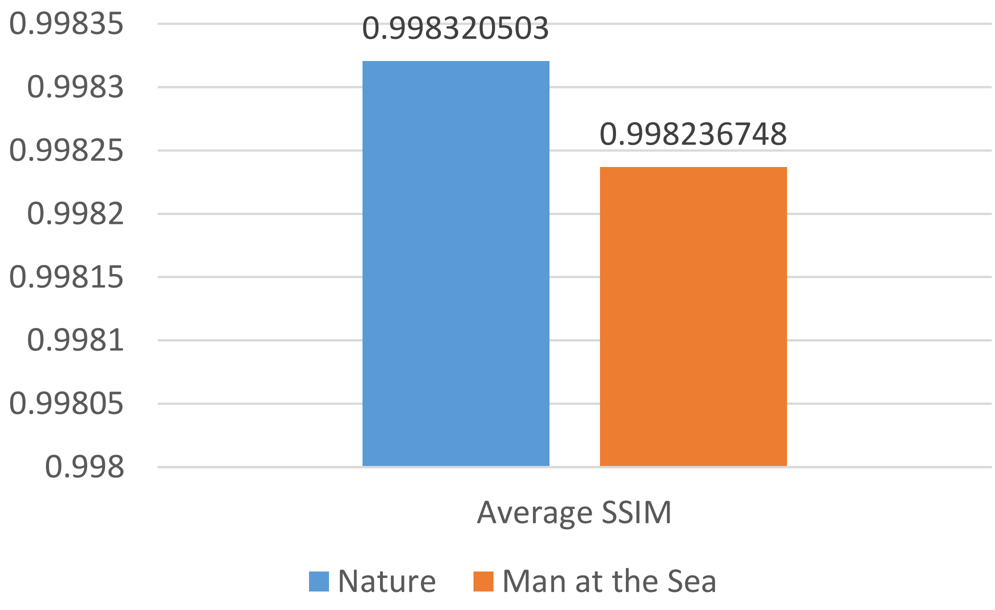
- Attacked Videos: To evaluate the robustness of the method against various attacks we attacked either a single frame or 10% of the frames of the video.

4. Evaluation

Measuring Fidelity Imperceptibility

To evaluate the fidelity and imperceptibility of the method we compute the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index Measure (SSIM).





4. Evaluation

AlgoLab
Algorithms Engineering Lab
Department of Computer Science & Engineering
University of Ioannina

Measuring Fidelity Imperceptibility

			Nature		Man at the Sea	
			Average PSNR	Average SSIM	Average PSNR	Average SSIM
Noise Attacked Frames	Gaussian	Single Attacked Frame Noise 0.5	99.85	1.00	99.85	1.00
		Multiple Attacked Frames 0.4	99.37	1.00	99.37	1.00
	Salt & Pepper	Single Attacked Frame Noise Salt & Pepper	99.50	0.99	99.49	0.99
		Multiple Attacked Frames Noise	94.04	0.91	93.92	0.92
Blur Attacked Frames	Gaussian	Single Attacked Frame (Kerner size 5)	99.50	1.00	99.52	1.00
		Multiple Attacked Frames (Kerner size 9)	94.32	1.00	93.76	0.99
	Average	Single Attacked Frame (Kerner size 3)	99.51	1.00	99.53	1.00
		Multiple Attacked Frames (Kerner size 7)	94.22	0.99	93.64	0.99
	Median	Single Attacked Frame (Kerner size 9)	99.48	1.00	99.48	1.00
		Multiple Attacked Frames Median (Kerner size 3)	94.91	1.00	94.57	1.00
Enhance	HEQ	Single Attacked Frame	99.42	1.00	99.41	1.00
		Multiple Attacked Frames	93.12	0.97	92.94	0.97
	Gamma	Single Attacked Frame 1.25	99.42	1.00	99.41	1.00
		Multiple Attacked Frames 2.5	93.08	0.96	92.95	0.95
Compression		Single Attacked Frame 50	99.58	1.00	99.55	1.00
		Multiple Attacked Frames 70	94.54	1.00	94.21	1.00
Crop		Single Attacked Frame Width 25%	99.47	1.00	99.46	1.00
		Multiple Attacked Frames Height 50%	93.36	0.95	93.22	0.95

AlgoLab
Algorithms Engineering Lab
Department of Computer Science & Engineering
University of Ioannina

Video Authenticity Verification (Frame-by-Frame)

			#UF	$\bar{S}$ Extract (%) [UF]	#AF	$\bar{S}$ Extract (%) [AF]	Total
Nature			125	0.9985	-	-	
Noise Attacked Videos	Gaussian	Single Attacked Frame	124	0.9985	1	1.0000	0.999
		Multiple Attacked Frames	113	0.9986	12	0.9974	0.998
	S&P	Single Attacked Frame	124	0.9985	1	0.2188	0.992
		Multiple Attacked Frames	113	0.9985	12	0.1563	0.914
Blur Attacked Videos	Gaussian	Single Attacked Frame	124	0.9985	1	0.2031	0.992
		Multiple Attacked Frames	113	0.9986	12	0.2188	0.921
	Average	Single Attacked Frame	124	0.9985	1	0.2344	0.992
		Multiple Attacked Frames	113	0.9985	12	0.2135	0.920
	Median	Single Attacked Frame	124	0.9985	1	0.1875	0.992
		Multiple Attacked Frames	113	0.9983	12	0.2031	0.919
Enhance Attacked Videos	HEQ	Single Attacked Frame	124	0.9985	1	0.9844	0.998
		Multiple Attacked Frames	113	0.9985	12	0.9831	0.997
	Gamma	Single Attacked Frame	124	0.9985	1	0.8906	0.998
		Multiple Attacked Frames	113	0.9983	12	0.7708	0.976
Compressed Videos		Single Attacked Frame (90)	124	0.9985	1	0.25	0.993
		Multiple Attacked Frames (70)	113	0.9985	12	0.1966	0.918
Cropped Videos		Single Attacked Frame (25)	124	0.9985	1	0.7656	0.997
		Multiple Attacked Frames (50)	113	0.9986	12	0.5456	0.953

4. Evaluation

Video Authenticity Verification (Frame-by-Frame)

			#UF	$\bar{S}$ Extract (%) [UF]	#AF	$\bar{S}$ Extract (%) [AF]	Total
Man at the Sea			125	1.0000	-	-	
Noise Attacked Videos	Gaussian	Single Attacked Frame	124	1.0000	1	1.0000	1.000
		Multiple Attacked Frames	113	1.0000	12	1.0000	1.000
	S&P	Single Attacked Frame	124	1.0000	1	0.1875	0.994
		Multiple Attacked Frames	113	1.0000	12	0.1758	0.918
Blur Attacked Videos	Gaussian	Single Attacked Frame	124	1.0000	1	0.1875	0.994
		Multiple Attacked Frames	113	1.0000	12	0.2031	0.920
	Average	Single Attacked Frame	124	1.0000	1	0.2344	0.994
		Multiple Attacked Frames	113	1.0000	12	0.2214	0.922
	Median	Single Attacked Frame	124	1.0000	1	0.1875	0.994
		Multiple Attacked Frames	113	1.0000	12	0.2305	0.923
Enhance Attacked Videos	HEQ	Single Attacked Frame	124	1.0000	1	0.9375	1.000
		Multiple Attacked Frames	113	1.0000	12	0.9818	0.998
	Gamma	Single Attacked Frame	124	1.0000	1	0.8281	0.999
		Multiple Attacked Frames	113	1.0000	12	0.6784	0.968
Compressed Videos		Single Attacked Frame (90)	124	1.0000	1	0.2188	0.994
		Multiple Attacked Frames (70)	113	1.0000	12	0.2161	0.922
Cropped Videos		Single Attacked Frame (25)	124	1.0000	1	0.7656	0.998
		Multiple Attacked Frames (50)	113	1.0000	12	0.5143	0.951

4. Evaluation

Video Authenticity Verification Results (Integrity Preservation Results)

			Nature	Man at the Sea
Watermarked Video			Hash of Frames and Hash Extracted from Audio Match	Hash of Frames and Hash Extracted from Audio Match
Noise Attacked Videos	Gaussian	Single Attacked Frame	Hashes do **not** Match	Hashes do **not** Match
		Multiple Attacked Frames	Hashes do **not** Match	Hashes do **not** Match
	Salt & Pepper	Single Attacked Frame	Hashes do **not** Match	Hashes do **not** Match
		Multiple Attacked Frames	Hashes do **not** Match	Hashes do **not** Match
Blur Attacked Videos	Gaussian	Single Attacked Frame	Hashes do **not** Match	Hashes do **not** Match
		Multiple Attacked Frames	Hashes do **not** Match	Hashes do **not** Match
	Average	Single Attacked Frame	Hashes do **not** Match	Hashes do **not** Match
		Multiple Attacked Frames	Hashes do **not** Match	Hashes do **not** Match
	Median	Single Attacked Frame	Hashes do **not** Match	Hashes do **not** Match
		Multiple Attacked Frames	Hashes do **not** Match	Hashes do **not** Match
Enhance Attacked Videos	Histogram Equalization	Single Attacked Frame	Hashes do **not** Match	Hashes do **not** Match
		Multiple Attacked Frames	Hashes do **not** Match	Hashes do **not** Match
	Gamma	Single Attacked Frame	Hashes do **not** Match	Hashes do **not** Match
		Multiple Attacked Frames	Hashes do **not** Match	Hashes do **not** Match
Compressed Videos		Single Attacked Frame	Hashes do **not** Match	Hashes do **not** Match
		Multiple Attacked Frames	Hashes do **not** Match	Hashes do **not** Match
Cropped Videos		Single Attacked Frame	Hashes do **not** Match	Hashes do **not** Match
		Multiple Attacked Frames	Hashes do **not** Match	Hashes do **not** Match

5.Conclusion

Potentials and Limitations

Potentials:

- ✓ Great quality of the watermarked video as the results of the PSNR and SSIM values show.
- ✓ Difficult for an attacker to accurately locate and extract the watermark with the 2-levels of frame partition (frame partition into blocks, blocks are partitioned into cells from the 2D representation of Self-Inverting Permutations of the integers and only certain of those cells are marked).
- ✓ Great average extraction rate allowing the owner to extracted the embedded watermark even when part of the frames are damaged.
- ✓ Can accurately detect tampering in the frames by comparing the hash of the frames and the embedded hash in the audio.

5.Conclusion

Potentials and Limitations

- Limitations:
  - The trade-off between robustness and imperceptibility. More robust watermarking creates artefacts in the frames, while keeping a watermark invisible involves minimizing the changes that happen.
  - The partition of frames. Partitioning in small and detailed areas (fine-grained blocking) the watermark does not affect the frame as much as adding the watermark into much larger areas (coarse-grained blocking) which spread the watermark information in the block and is not embedded precisely. But fine-grained blocking requires higher computational cost.
  - When the all the frames of video are damaged by an attack that the algorithm proves to not be very robust against (for example blurring attacks) the ability to extract the watermark is limited.

5.Conclusion

Future Research

- Future Research:
  - Further research should be done to improve the robustness against more attacks.
  - The method proved to be extremely successful in identifying tampering in the frames. Future research could focus on detecting and localizing the area where the tampering happened.
  - Given the promising results of the method potential expansions for real time watermarking should be investigated, for example watermarking of live streaming videos.