Arabian Journal for Science and Engineering
https://doi.org/10.1007/s13369-025-10027-x

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

Blockchain-Based Computationally Efficient Procedure for Multimedia Security Utilizing Bit Plane Slicing and Wavelet Transform

Noha AbdElkareem[1] · **Mazen Selim**[1] · **Ahmed Shalaby**[1]

Received: 27 March 2024 / Accepted: 10 February 2025

Abstract
With the rapid growth of multimedia data in various domains, ensuring the security and integrity of digital content has become a progressively challenging task. Research on multimedia integrity revolves around detection techniques. This paper introduces a new approach to securing the integrity of multimedia data based on the decentralized attributes inherent in blockchain (BC) technology. Our approach enhances the efficiency of hash block calculation for multimedia content (MC) by utilizing bit plane slicing and the first and second decomposition levels of the discrete wavelet transform (DWT). Our evaluation shows a substantial reduction in computational time, all while upholding the integrity of the MC.

Keywords Blockchain · Bit plane slicing · Discrete wavelet transform · Multimedia security

1 Introduction

Multimedia, including video, audio, images, and other digital content, has become integral to our daily lives. However, as the volume of MC shared online continues to grow, securing MC from unauthorized access, modification, and distribution has emerged as a critical concern in the digital age. One prominent challenge is the pervasive threat of digital piracy, where unauthorized distribution poses a significant risk to the economic sustainability of content creators and distributors. The movie and TV industry are estimated to lose between $40 and $97.1 billion annually due to digital piracy. Over 230 billion views of pirated videos occur annually [1]. Additionally, the advent of deepfake technology introduces a disconcerting dimension, enabling the creation of highly realistic fake videos and audio recordings that can be maliciously exploited. Furthermore, the covert embedding of information within multimedia files through steganography further complicates security efforts, posing challenges for detection. Moreover, the global nature of online platforms raises jurisdictional and legal complexities in enforcing MC security measures effectively. Securing MC involves implementing a combination of technical and procedural measures to maintain the three cornerstone security properties: confidentiality, integrity, and availability [2]. *Confidentiality* means to prevent untrusted users from reading, writing, or observing MC to be protected. Encryption is utilized to guarantee confidentiality. Encrypting multimedia files ensures that even if unauthorized individuals gain access to the files, they won't be able to view or use them without the decryption key. In addition, access control mechanisms such as usernames, passwords, and multi-factor authentication are implemented to provide robust access controls and restrict who can access MC. *Integrity* means verifying that MC has not been modified. Digital signature and digital watermarking are commonly utilized to track the source of the content and detect tampering or unauthorized distribution. *Availability* ensures the reliability and trustworthiness of accessing MC. Monitoring and auditing systems are deployed to track access and usage of MC and detect any unusual activities or unauthorized access and maintain multimedia distribution securely and efficiently.

Multimedia integrity solutions have emerged as an important research topic. Solutions like digital signature, watermarking, and tampering detection utilizing artificial intelligence algorithms have been proposed to address the integrity problem. Nevertheless, the majority of the proposed solutions focus on tampering detection, not protection. To protect multimedia, BC has been proposed recently. BC enhances security and helps prevent fraud and unauthorized activity [3, 4]. A BC [5] is a shared, immutable ledger based on

✉ Noha AbdElkareem
noha.eldemerdash@fci.bu.edu.eg

[1] Computer Science Department, Faculty of Computers and Artificial Intelligence, Benha University, Banha, Egypt

Published online: 19 March 2025

cryptography and distributed systems. It facilitates transaction recording and asset tracking. It offers several advantages in the context of multimedia integrity. Firstly, BC technology provides transparency by allowing all parties to view the same information, ensuring the authenticity and integrity of MC. Secondly, the decentralized nature of BC ensures data integrity, as records cannot be modified without consensus. This can help prevent unauthorized access, modification, and distribution of MC. Thirdly, BC technology is highly secure and resistant to tampering, which can help prevent unauthorized access, modification, and distribution of MC. Finally, BC technology allows tracking, which can help prevent unauthorized access, modification, and distribution. In our work, we propose a method based on BC concepts to assure the integrity, where multimedia resources can be stored in a secure and tamper-proof manner. We propose to utilize DWT [6] and bit plane slicing [7] to extract compact feature details about the multimedia resource. Then this information is formed in the shape of BC transactions. Next, the hash function is applied to transactions; finally, transactions are chained together, forming a BC.

Our proposed method preserves multimedia integrity and significantly reduces time consumption(TC), leading to an applicable solution for utilizing BC for various industries and applications. Below we list the key contributions of our work:

- We propose a novel method to secure multimedia resources based on BC technology, DWT, and bit plane slicing.
- We investigate the potential of BC for securing MC, deviating from the traditional approach of hashing all data within the frame. Instead, we adopt a more resource-efficient strategy by focusing on essential information using DWT and bit plane slicing. This targeted approach not only improves the effectiveness of our solution but also aligns seamlessly with the decentralized and secure characteristics of BC. Consequently, it contributes to a more streamlined and cost-effective method of ensuring the security of MC.
- We implement, evaluate, and analyze the performance of the proposed method to show the pros and cons.

The rest of the paper is structured as follows: Sect. 2 provides an overview of various techniques for multimedia security, including the nature of BC and hash function. It also introduces multimedia compression and explores the challenges for multimedia security. Section 3 explores the motivation behind our proposed method, shedding light on the challenges faced in multimedia security. Section 4 conducts a comprehensive review of existing research, examining various methodologies employed in securing MC. Moving on to Sect. 5, we present our innovative approach, serving as the central core of our study. This section provides a detailed breakdown of our proposed method. Section 6 focuses on the implementation and evaluation, showcasing tangible evidence of the effectiveness of our approach in securing MC. This part is augmented by visual representations, statistical analysis, and qualitative assessments for a thorough evaluation of our method's performance. Finally, Sect. 7 concludes the paper by summarizing the innovative aspects of our approach, offering a concise overview of the study's key contributions.

2 Background

As the digital landscape continues to evolve, the demand for robust multimedia security solutions has become increasingly crucial due to the rising volumes of sensitive and personal data shared through multimedia channels, social media, file-sharing applications, and cloud computing. In this section, we provide an overview of multimedia security and multimedia compression, highlighting the challenges and solutions.

2.1 Multimedia Security

The risk of data breaches and unauthorized access to MC has significantly escalated. According to DeepMedia (Reuters), an estimated 500,000 video and voice deepfakes were distributed across global social media platforms in 2023. The count of deepfake videos on the internet has witnessed a twofold increase since 2018, reaching 14,678 videos in 2021, as reported by Deeptrace [8]. The proliferation of data sharing has become one of the primary challenges in multimedia security. To address this issue, numerous techniques have been proposed for verifying the origin and authenticity of multimedia data as follows:

- Digital signatures are a way to verify the authenticity and integrity of MC by using public key cryptography. This technique adds a digital signature to the video that can be used to verify that the video has not been tampered with or altered [9].
- Watermarking techniques embed a unique identifier into the MC that can be used to verify its authenticity. This technique is helpful in protecting intellectual property rights because it enables the multimedia owner to track its distribution [10].
- Sophisticated methods employ machine learning algorithms for the detection of tampering or modification in MC. This approach is useful in detecting subtle changes that may not be noticeable to the human eye [11].

- Cryptographic techniques use encryption algorithms to protect the privacy and confidentiality of MC. This technique is used to ensure that only authorized users can access the MC and prevent unauthorized access or manipulation [12].

Overall, there are various techniques to protect the integrity and authenticity of multimedia. These techniques can be used together or separately, and each technique has its own strengths and limitations. The choice of technique depends on the specific requirements and constraints of the application. Recently, blockchain (BC) is proposed. BC technology can provide a secure and tamper-proof way to store and verify original, unaltered media content. BC consists of a chain of blocks, where each block contains a list of transactions or data and a reference to the previous block. The entire chain is cryptographically linked, ensuring that any change to a block will be evident in subsequent blocks, making it tamper resistant [13].

With further elaboration, BC uses hash functions as they play a vital role in ensuring the integrity and security of the data stored in blocks. Each block contains a hash value representing the data it holds, and this hash value is based on the content of the block, including the previous block's hash value. If any part of the data in a block is altered, its hash value will change, breaking the link to the next block and thus making it evident that tampering has occurred. A hash utilizes a mathematical algorithm to process input "message" of any size and produce a fixed-size string of characters known as a hash value or digest. Hash functions have several important properties:

- Deterministic: For a given input, the hash function always produces the same hash value.
- Fast computation: Hash functions are designed to be efficient.
- Irreversibility: It should be computationally infeasible to reverse-engineer the original input from its hash value.
- Avalanche effect: A small change in the input should produce a significantly different hash value.

The main goal of a hash function is to efficiently and rapidly map data of arbitrary length to a fixed-size output, typically generating a unique representation of the input data. Several algorithms, such as MD-5, SHA-1, SHA-2, SHA-3, and SHA-256, are commonly used to calculate hash values [14]. SHA-256 (Secure Hash Algorithm 256-bit) offers numerous benefits compared to MD-5 (Message Digest Algorithm 5). One primary advantage lies in the significantly larger bit size of SHA-256, resulting in a more extensive and more secure hash value. This increased hash size makes it computationally impractical for attackers to carry out collision attacks, where distinct inputs yield the same hash output. Moreover, SHA-256 is deemed more resilient to cryptographic vulnerabilities compared to MD-5. The use of SHA-256 in BC technology further enhances its security strength, making it a more reliable choice for cryptographic applications and data integrity verification. In summary, hash functions help maintain the integrity of data within BC by providing a secure and efficient way to verify the authenticity of the data and detect any changes or tampering attempts.

2.2 Multimedia Compression

Multimedia compression plays a pivotal role in addressing the challenges of escalating computational complexity of security protocols such as encryption, digital signatures, and hash functions. This paper proposes leveraging image processing techniques, specifically bit plane slicing and wavelet transform, to streamline security operations and reduce the computational burden in BC systems while preserving efficiency and reliability.

2.2.1 Bit Plane Slicing

Bit plane slicing analyzes and processes image data by separating each pixel binary representation into separate layers, or "planes." The Most Significant Bits (MSBs), found in higher-order planes (e.g., plane 7), primarily define the image's visual quality, while the Least Significant Bits (LSBs), found in lower-order planes (e.g., plane 0), typically capture finer details or noise.

Mathematically, the intensity $I(x, y)$ of a pixel at position (x, y) can be expressed as:

$$I(x, y) = \sum_{k=0}^{n-1} b_k(x, y) \cdot 2^k$$

where $b_k(x, y)$ is the binary bit value (0 or 1) at the k-th bit plane. For $k = 0$, $b_0(x, y)$ represents the LSB.

For $k = n - 1$, $b_{n-1}(x, y)$ represents the MSB.

By extracting specific bit planes, such as the MSB planes, we can focus on essential image information while reducing redundancy and computational load. This technique is particularly valuable in applications such as image compression, watermarking, feature extraction, and multimedia security, as it facilitates the efficient processing and storage of high-priority data while maintaining essential visual or functional characteristics.

2.2.2 Discrete Wavelet Transform

The discrete wavelet transform (DWT) is a versatile and robust technique for decomposing signals or images into distinct frequency sub-bands, facilitating multi-resolution analysis. Specifically, it decomposes the input into one low-frequency component, **LL (Low-Low)**, and three high-frequency components: **LH (Low-High)**, **HL (High-Low)**, and **HH (High-High)**, which correspond to horizontal, vertical, and diagonal details, respectively. Each computed value represents the intensity of a pixel within a specific sub-region of the transformed image. Among these, the LL sub-band retains the primary structural and visual characteristics of the image, serving as a coarse approximation. This makes it critical for applications like security enhancement and redundancy reduction.

Building on this foundation, the second-level decomposition of the DWT recursively applies the transform to the LL sub-band from the first level, further dividing it into four new sub-bands: LL2, HL2, LH2, and HH2. The LL2 sub-band captures coarser structural information, while HL2, LH2, and HH2 represent finer horizontal, vertical, and diagonal details. By capturing image features at progressively finer scales while preserving the coarse approximation, second-level decomposition enhances multi-resolution analysis. This hierarchical representation is especially beneficial for applications such as image compression, where redundancy is minimized, and security tasks, where selective manipulation or analysis of specific details is required.

3 Motivation

Ensuring the integrity of multimedia files, including images, videos, and audio recordings, is crucial to guarantee their accuracy, consistency, and reliability. Nevertheless, safeguarding MC against malicious alterations poses a substantial research challenge. Current research in multimedia integrity mainly focuses on detection techniques, such as hash-based methods and digital signatures. Several challenges arise in this field, with one notable obstacle being the increasing cost associated with verifying the authenticity and integrity of multimedia files. As technology evolves and multimedia files become more advanced, the verification process becomes more time-consuming and resource intensive.

Another significant challenge is multimedia attacks, including the emergence of deepfake [15]. Deepfakes utilize AI to generate realistic but fake MC. The rapid advancement of deepfake technology and its potential for misuse is an additional challenge for multimedia security. Deepfakes can compromise the security of previous techniques introduced in the background section such as digital signatures and watermarks by creating false or misleading videos and images, which can harm reputations and spread false information. Therefore, individuals, organizations, and governments must be aware of the risks associated with deepfakes and take necessary measures to counteract them. However, one of the main challenges with deepfakes lies in the difficulty of detecting them.

BC has the potential to address the deepfake threat. In BC, the hash of each block acts as a digital fingerprint, ensuring the integrity of the original content. In the case of deepfake detection, a decentralized network of nodes can be employed to verify the authenticity of MC. As deepfake detection algorithms evolve, BC can serve as a tamper-resistant repository, making it significantly challenging for malicious actors to manipulate content without leaving a trace. This not only enhances the credibility of multimedia data but also provides a practical solution to the growing threat of deceptive and malicious deepfake content.

Furthermore, the emergence of Web3, also known as Web3.0, represents the third generation of the World Wide Web. It aims for a more decentralized, intelligent, and user-centric internet. Unlike its predecessors, it incorporates decentralized technologies like BC to reduce dependence on central authorities. This shift envisions increased user control over personal data, heightened privacy measures, and the development of Decentralized Applications (DApps) and platforms. Web3 differs from traditional web architectures by utilizing BC as a decentralized ledger. Data are distributed across a network of nodes, each maintaining a copy of the entire BC for redundancy and tamper resistance. Web3 employs peer-to-peer networks, enabling direct communication between nodes without intermediaries. It often utilizes decentralized storage systems, such as the InterPlanetary File System (IPFS), breaking files into smaller chunks distributed across nodes for efficient retrieval and redundancy. The integration of BC in Web3 addresses challenges posed by deepfakes by providing a secure and reliable platform for verifying the authenticity and integrity of MC. The evolution to Web3 has heightened the demand for innovative security solutions in Web3 applications. A notable challenge is ensuring the security and verification of MC within a decentralized Web3 environment, distributed across multiple nodes. To overcome this challenge, it is crucial to leverage BC technology to create verifiable and secure MC. Various methods, including BC-based authentication, content protection, and decentralized storage, can contribute.

In conclusion, recent technological advancements, such as Web3 and deepfakes, have added complexity to the security landscape of multimedia. This has intensified the need for creative approaches to secure and verify MC while reducing the required time in various environments, including decentralized web3 environments. *Our goal is to present a well-balanced approach that ensures integrity without*

imposing significant limitations while maintaining a reasonable computational cost.

4 Related Work

Recently, different techniques based on BC technology are proposed for assuring the integrity of multimedia. These techniques can be divided into schemes as follow

4.1 Scheme Based on Splitting the Video into Segments or Frames

It involves calculating the hash of each segment, encrypting the hash values, storing these hash values in the BC, and comparing the calculated hash values with the stored ones to check for tampering.

In [16], the authors proposed an approach that uses a combination of BC and cloud services to secure digital video from unauthorized modification. The video is segmented into frames, and a hash for each frame is calculated using the SHA-2 algorithm. The hash of each frame is used to form a BC, with each block containing the hash sum of the previous frame. This approach is well suited for video surveillance systems because it verifies the integrity of the video by calculating the hash for each frame. However, it is a time-consuming technique due to the increased computational requirements of the hash calculation process for each frame.

In [17], BlockSee is proposed to verify the integrity of video surveillance footage based on BC technology. It splits the video into frames and calculates the hash for each frame, which is then sent to the BC. The frames are segmented into foreground and background, with the background being fixed so that any changes to the scene can be detected. The camera settings are verified by extracting features from the background using the SIFT algorithm, which are then sent to the BC. The movement of video segments is referred to as a BlockSee transaction, which consists of the hash of the video segment, a timestamp, the extracted background features, and the camera ID. The video transactions are grouped into blocks, and the blocks are linked together to form the BlockSee BC.

In [18], the authors proposed to split the video into segments, with each segment consisting of a few minutes. The hash for each segment is calculated using the SHA-256 algorithm and a message authentication code (MAC) using HMAC with the HMAC key (dk). Each video segment has a unique value generated randomly, which is referred to as dk. To prevent attackers from being able to edit the video without leaving any marks, the key is encrypted using Elliptic Curve Cryptography (ECC) to make the system more secure. The video MAC is encrypted using dk, and the output of the encryption is referred to as the video integrity code (VIC). A randomized hashing is applied to dk to generate the block HMAC key (bk). The output from data encryption and keys is sorted into blocks and linked together to form a BC. Although this system is more secure than other conventional methods, the average time for hashing is 8 ms longer than other methods that do not use BC technology.

In [19], an approach for IoT devices utilizes BC technology. The encoder compresses the raw video captured by the camera through the generation of I and B frames. The video is divided into segments, each potentially containing multiple I-frames and several B frames, but consistently commencing with an I-frame. These segments have an average duration of approximately 3 s. The SHA-3 algorithm is employed to apply a hash function to each segment, calculating hash values that are subsequently stored in the BC. Extensive testing of the system has been conducted on both a Raspberry Pi and the Hyperledger platform.

Other approaches aim to reduce the time of computation by hashing only selected frames, such as I-frames, using encoding techniques like H.264 or MPEG. In [20], a method aimed at verifying the integrity of videos on IoT devices through BC technology is presented for forensic investigation purposes. This technique involves calculating the hash of video frames within the IoT device before their transmission to a remote base station. To ensure secure transmission, the hash is sent over a TCP-based connection. Subsequently, this hash is stored on multiple nodes on a permissioned BC platform, emphasizing a reduction in computation time. The video undergoes segmentation into frames, followed by conversion into string values. A hash is computed for each string using the MD-5 hash function. In order to optimize computation time, the video is encoded using H.264 or MPEG encoding, and only a selected number of frames, specifically I-frames, undergo hashing.

4.2 Schemes Use Cryptography Algorithms

The aim is to ensure the secure communication of hash values in the BC. In [21], a technique employing BC technology and ECC is introduced to ensure the authenticity of the video. The video is partitioned into segments, with each segment spanning a few minutes. The Elliptic Curve Digital Signature Algorithm (ECDSA) is applied to compute a digital signature for each segment. These signatures are then stored in interconnected blocks, forming a BC. Each block encompasses essential information, including the signature of the video segment, the public key associated with the video segment signature, the signature of the preceding block, the public key related to the previous block signature, the block index, and the path of the video file.

In [22], a technique for assuring the integrity of surveillance video is proposed, consisting of two main methods:

Data Integrity Function (DIF) and Integrity Validation Function (IVF). The video is segmented into smaller videos recorded every few minutes. In the DIF, the hash value of each video segment is calculated using randomized hashing and encrypted with a key using the ECC algorithm. This ensures the integrity of the video. The IVF is responsible for validating the integrity of the video. To validate the integrity of the video, the hash value is calculated for the target video segment and the encrypted hash of stored video segments is decrypted using the ECC private key. The target video hash is compared with decrypted hash values to determine if the video has been tampered with. This technique is more robust against numerous attacks, but it takes more time to use the randomization method.

In [23], authors proposed a system based on BC technology for managing video surveillance. An IP camera records the video and then transferred to a BC Video Management (BVM) application. The video is encrypted using the Advanced Encryption Standard (AES) algorithm and sent to an IPFS node within the BVM. A hash of the video is then generated. The decryption key of the video is managed by the private database of the BC, ensuring that the internal manager cannot leak it to unauthorized users. Additionally, the internal administrator can safely export and manage videos by exporting the license generated in the BC to a Digital Rights Management (DRM)-applied video player.

4.3 Schemes Use Off-BC Storage to Store Video Data Securely

The SePriS system in [27] employs BC technology to safeguard the integrity and authenticity of surveillance video files stored in off-BC storage linked to BC nodes through smart contracts. The system introduces the DAB schema for encrypting video frames, encompassing four primary stages. These stages involve utilizing the Discrete Cosine Transform (DCT) to convert spatial data into frequency data, employing a quantizer to reduce the amount of information needed to represent it, utilizing the AES to encrypt the quantized data to ensure confidentiality and integrity, and incorporating a block shuffler (BS) for rearranging encrypted data to add an additional layer of security. The DAB schema is meticulously crafted to establish a secure method for encrypting video frames, thereby enhancing the overall integrity and confidentiality of the video content.

The approach described in [26] aims to preserve the reliable integrity of a video using BC technology. This is achieved through the implementation of two mechanisms. In the first mechanism, software characteristics extracted from digital camera devices and video editing tools are integrated into the BC to verify the integrity of the digital video file. This process includes comparing the extracted characteristics with signature databases for video editing tools and digital camera devices, resulting in the determination of the video file's originality based on matched signatures. The second mechanism involves analyzing the modification history of a tampered video file by searching the comparison results stored in the BC system. The index used for searching is the hash value of the digital video, ensuring efficient and reliable retrieval.

4.4 Schemes for Assuring Integrity of Image and Text Files

In [24], ImageChain is proposed, an approach based on cryptography that connects a group of images by calculating a hash function for each image using the SHA-256 algorithm. Instead of using blocks to form a BC, this method inserts the hash of each image into the image itself, forming an ImageChain. The hash of each image includes the hash of the previous image, allowing the creation of a chain of hashes. This approach eliminates the need for additional files or software, as the images themselves are used to verify the integrity of the images. It also supports any image format.

In [25], a double steganography approach is proposed. It combines the Interplanetary File System (IPFS) and BC technology. Secret information is encrypted using the RSA algorithm and then converted into binary form, which is hidden within an image using the Least Significant Bit (LSB) technique. The resulting image is referred to as a "stego" image. The hash of the stego image is then embedded in text files using the whitespace method. These files are sent through the BC network using covert channels, making it difficult for unauthorized users to differentiate between normal transactions and those that include secret information. This system is more robust against the detection of secret information compared to other methods.

Table 1 concludes a comparison of the related work. In summary, the initial schemes exhibit limitations in complexity and TC due to hashing for all frames. Conversely, schemes aiming to minimize complexity by selecting specific frames face the issue of potential tampering with missing frames or video parts, thereby compromising security. Nevertheless, the other schemes involve trade-offs related to complexity, TC, and the level of integrity assurance they provide.

5 Proposed Work

Our objective is to introduce a balanced approach that ensures integrity without limitations while maintaining reasonable TC compared to other methods. We have successfully achieved a TC reduction of around 70% compared to alternative methods. In this section, we detail our proposed approaches, specifically introducing a method that incorporates bit plane slicing and DWT decomposition to extract key features from video frames in order to minimize the TC.

Table 1 Related work comparative analysis

Reference	Type of multimedia	Techniques used	Advantages	Drawbacks
[16]	Video	It splits the video into frames and calculates the hash sum for each frame using the SHA-2 algorithm to form BC	It is suitable for the video surveillance system. It protects the integrity of video files	The hash calculation process may become time-consuming for high-resolution videos due to increased computational demands
[24]	Image	It calculates a hash function for each image using the SHA-256 algorithm. Instead of using blocks to form a BC, this method inserts the hash of each image into the image itself, forming an ImageChain	The pictures are not stored inside the blocks, thereby removing the necessity of a separate ledger. It does not require additional files, except for the images themselves	More time is required due to the image hash process
[17]	Video	It splits the video into frames, calculates the hash for each frame, and sends it to the BC. The frames are segmented into foreground and background, and features are extracted from the background using the SIFT algorithm, which are then sent to the BC	It ensures the integrity of surveillance videos, preventing tampering or unauthorized modifications	The system's performance is influenced by both complexity and TC
[18]	Video	It splits the video into segments, each lasting a few minutes. The hash for each segment is calculated using the SHA-256 algorithm, and a message authentication code (MAC) is generated using HMAC with the HMAC key	It provides significantly greater robustness against tampering detection compared to other conventional methods	Increased computational overhead: Splitting the video into segments and applying cryptographic algorithms to each segment adds computational load. The storage of video segments, hashes, encrypted keys, and BC data may also demand more storage space
[25]	Image and text files	Secret information is encrypted using the RSA algorithm and concealed within an image through the Least Significant Bit (LSB) technique, resulting in what is known as a "stego image." The hash of the stego image is then embedded in text files using the whitespace method. These files are transmitted through the BC network using covert channels	The more robust the anti-detection performance, the more challenging it becomes for unauthorized or malicious parties to detect the secret information	When message concealment is guaranteed, the real-time performance of the message is relatively low due to the certain chronological order inherent in the original plain text files, limiting the scope of application for the model
[23]	Video	It encrypts and stores the video, creates a license within the BC, and then exports the video. The internal administrator can securely manage and export videos by utilizing the license generated in the BC	It securely manages videos from both external persons and internal administrators	Storing video surveillance data on the BC can require substantial storage space, and the computational intensity increases when encrypting and decrypting videos using the AES algorithm

Table 1 continued

Reference	Type of multimedia	Techniques used	Advantages	Drawbacks
[26]	Video	It consists of two mechanisms. The first mechanism involves extracting software characteristics from the digital camera device and video editing tools, then inserting them into the BC to ensure the integrity of the digital video file. The second mechanism involves analyzing the conversion history of a tampered video file by searching the saved comparison results in the BC system	The proposed mechanism minimizes the amount of data stored in the BC system, thereby enhancing the reliability of integrity analysis results	The approach may not provide a strong defense against sophisticated attacks or advanced manipulation techniques. Furthermore, it entails increased computational overhead, as it requires extracting software characteristics from digital camera devices and video editing tools, as well as analyzing conversion histories
[21]	Video	The video is divided into segments, and a digital signature is calculated for each segment using the Elliptic Curve Digital Signature Algorithm (ECDSA). These signatures are then stored in blocks that are linked together to form a BC	It can detect any type of forgery and applies to any kind of video	"Storing the signatures of each video segment in blocks within the BC can result in increased storage and computational requirements
[20]	Video	The video is segmented into frames, converted into string values, and a hash for each string is calculated using the MD-5 hash function. To reduce computation time, the video is encoded using H.264 or MPEG encoding, and only a select number of frames, specifically I-frames, are hashed	It reduces computation time and eliminates the need to store the hash of each frame	It may miss potential tampering or alterations occurring in non-selected frames
[19]	Video	The video is split into segments, each beginning with an I-frame and having an average duration of around 3 s. The hash function is applied to each segment to calculate hash values, which are then stored in the BC	The hash function does not have to wait for the video to be complete. With this approach, the video is instantly sent from the IoT device, enabling real-time streaming	Reduced granularity of integrity verification
[27]	Video	It proposes a Secure and Privacy-preserving Stored Surveillance Video Sharing (SePriS) mechanism for authorized users/nodes. The mechanism is based on smart contracts, BC, and the enciphering of video frames using DAB-a method developed from discrete cosine transform (DCT), AES, and BS algorithms	It ensures secure and privacy-aware surveillance practices during the sharing or accessing of stored surveillance videos	It does not provide complete secrecy for authorized users
[22]	Video	The video is segmented into smaller videos recorded every few minutes, and the hash value of each video segment is calculated using randomized hashing. Subsequently, the hash values are encrypted with a key using the ECC algorithm	The proposed method is more robust against various attacks and offers a higher level of security	The use of the randomization method results in a longer processing time

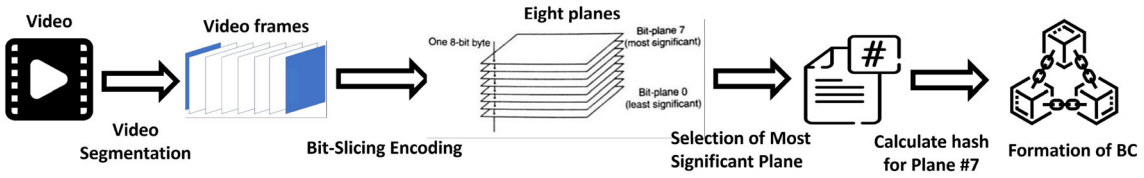

Fig. 1 Bit plane slicing approach

Subsequently, we incorporate decentralized BC concepts to preserve multimedia resources. Our approach enables efficient calculation while ensuring multimedia integrity.

5.1 Bit Plane Slicing Approach

Figure 1 summarizes the bit plane slicing approach where the approach steps can be described as follows:

1. Video Segmentation: The video is split into frames for efficient processing. Given a video V consisting of n frames, each frame can be denoted as F_i where i is the index of the frame, and $F_i \in R^{H\times W}$, representing the frame as a matrix of pixel intensities with height H and width W. V= $\{F_1, F_2, F_3, \ldots, F_n\}$ Each frame F_i is independently processed in subsequent steps, allowing parallel and efficient operations.
2. Bit-Slicing Encoding: Next, we apply the bit-slicing technique to decompose the frame into its bit planes. The bit-slicing technique is useful for efficiently extracting specific bit planes from each pixel's intensity value. For a pixel intensity value $I(x, y)$ in frame F_i, which lies within the range [0–255] (i.e., 8-bit grayscale), the binary representation is: $I(x, y) = b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$

$$b_k^{(i)}(x, y) = \left\lfloor \frac{I(x, y)}{2^k} \right\rfloor \bmod 2$$

   where $b_k^{(i)}(x, y)$ represents the k-th bit plane for the pixel (x, y)in frame F_i. This operation is performed for each pixel in the frame, resulting in 8-bit planes for each frame $\{ b_0^{(i)}(x, y),\ b_1^{(i)}(x, y), \ldots,\ b_7^{(i)}(x, y)\}$.
3. Selection of Most Significant Plane: After obtaining the eight-bit planes for each frame, the next step is to select the MSB plane. The MSB plane $b_7^{(i)}(x, y)$ contains the fundamental information regarding the structure of the image and can be used for integrity verification. Thus, for each frame F_i, the most significant bit plane $b_7^i(x, y)$ is chosen. Mathematically, we refer the selection as: $b_{\text{MSB}}^i(x, y) = b_7^i(x, y)$. This plane is then used for subsequent hash calculations and integrity checks.
4. Calculation of Hash Sum: To ensure the integrity of the selected plane and detect any modifications or tampering attempts, we calculate the hash sum of the most significant bit plane $b_{\text{MSB}}^{(i)}$ along with its adjacent frames. Let $\text{H}(b_{\text{MSB}}^{(i)})$ denote the cryptographic hash of the selected plane for frame F_i. The hash sum is computed by concatenating the current plane with the hash of the previous plane and then applying a cryptographic hash operation:

$$H_{\text{sum}}^{(i)} = \text{SHA} - 256(b_{\text{MSB}}^{(i)} || H\left(b_{\text{MSB}}^{(i-1)}\right))$$

   This approach ensures that both the current and preceding bit planes are considered for tampering detection. The outcome is a distinctive hash for the frame, effectively encapsulating the integrity of the bit plane in relation to its temporal context.
5. Formation of BC: Finally, the calculated hash sum for each frame is integrated into a BC structure to ensure tamper-proof verification and establish a secure, chronological chain of frames. Each BC block B_i contains the following elements:
   - $H_{\text{sum}}^{(i)}$: the hash sum of the selected plane and previous frame's plane, serving as unique digital fingerprints
   - T_i: the timestamp of frame F_i to establish chronological order and ensure integrity.
   - ID_i: the unique frame ID of F_i to provide distinct identification for each frame

   The BC block B_i contains: B_i=$\{H_{\text{sum}}^{(i)}\ T_i, \text{ID}_i\}$ By chaining these blocks together, as illustrated in Fig. 2, we form the video BC. This chain ensures a tamper-proof digital fingerprint for each frame, enabling efficient and secure video verification.

5.2 Discrete Wavelet Transform Approach

Another approach to minimize hash calculation is to use the DWT. Figure 3 presents the steps of the DWT approach using first-level decomposition. The steps are as follows:

1. Video Segmentation: Similar to bit-slicing approach, the video is segmented into individual frames F_i.
2. Encoding Each Frame into Four Sub-bands Using First-Level Decomposition of DWT: Next, we apply the first-level DWT to each frame F_i, which decomposes the frame into four sub-bands: LL, LH, HL, and HH. The four sub-bands are calculated as follows: Let $I(x, y)$ denote the intensity of the pixel at location (x, y) in frame F_i. *

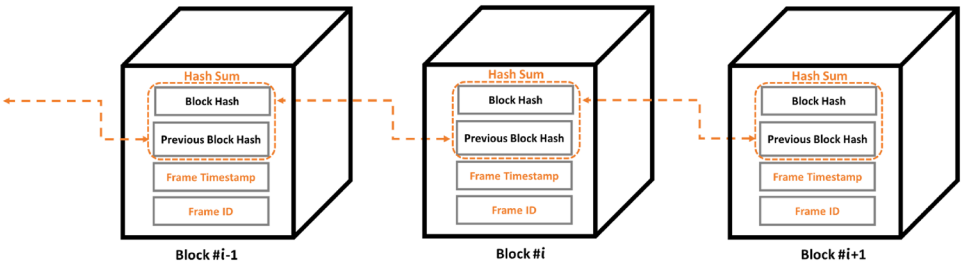

Fig. 2 Blockchain structure, the BC block B_i contains: B_i={$H_{\text{sum}}^{(i)}$ T_i, ID_i}

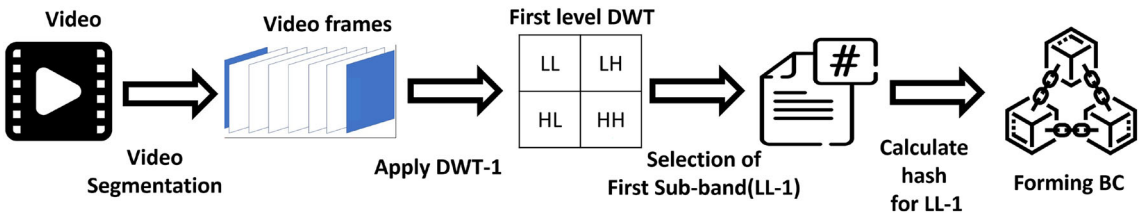

Fig. 3 First-level decomposition discrete wavelet transform (DWT) approach

LL (Approximation): Represents the average intensity of all four pixels, capturing the coarse approximation of the image.

$$LL = \frac{I(x,y) + I(x+1,y) + I(x,y+1) + I(x+1,y+1)}{4}$$

* **LH (Vertical Detail)**: Captures vertical differences between pixels.

$$LH = \frac{I(x,y) - I(x+1,y) + I(x,y+1) - I(x+1,y+1)}{4}$$

* **HL (Horizontal Detail):** Reflects horizontal differences between pixel values.

$$HL = \frac{I(x,y) + I(x+1,y) - I(x,y+1) - I(x+1,y+1)}{4}$$

* **HH (Diagonal Detail)**: Captures diagonal differences between pixels.

$$HH = \frac{I(x,y) - I(x+1,y) - I(x,y+1) + I(x+1,y+1)}{4}$$

Each of these operations yields one of the four sub-bands LL, LH, HL, and HH, which represent different frequency components of the frame.

3. Selection of First Sub-band: After obtaining the four sub-bands for each frame, the next step is to select the first sub-band, LL, which contains the fundamental information of the frame and is used for integrity verification. This selection is represented mathematically as: $b_{\text{selected}}^{(i)}$ = LL. This sub-band LL is then used for subsequent hash calculations.
4. Calculation of Hash Sum: we compute the hash sum of the selected LL sub-band, along with its adjacent frames. Let H($b_{\text{selected}}^{(i)}$) represent the cryptographic hash of the selected sub-band for F_i. The hash sum is computed by concatenating the current sub-band and the hash of the previous frame's sub-band, followed by a cryptographic hash operation:

$$H_{\text{sum}}^{(i)} = SHA - 256(b_{\text{selected}}^{(i)} || H\left(b_{\text{selected}}^{(i-1)}\right))$$

This ensures that both the current and previous sub-bands are considered when detecting tampering. The resulting hash represents the integrity of the selected sub-band in relation to its temporal context.

5. Formation of BC: Similar to the bit-slicing approach, a BC is constructed by chaining blocks. Each block contains the calculated hash sum of the selected LL sub-band($H_{\text{sum}}^{(i)}$), the frame's timestamp(T_i), and its unique frame ID (ID_i). This can be mathematically represented as: B_i={$H_{\text{sum}}^{(i)}$ T_i, ID_i} The chaining of these blocks ensures a secure, tamper-proof structure for video integrity verification.

To achieve further simplification and data reduction, we implemented the second-level decomposition of DWT. The second-level decomposition involves taking the output of the first-level decomposition and applying the DWT again to further break down the information into finer details. In our approach, the DWT is applied to the first sub-band output resulting from the first-level decomposition.

While our techniques were originally proposed for visual multimedia (videos and images), they can be extended to secure other forms of multimedia. We have adapted our approach to enhance security for both audio and textual data. In the case of text files, our process entails an initial transformation of the text into PDF format, followed by the conversion of the PDF into images. Subsequently, we apply our proposed techniques to secure the textual content within these images. Similarly, for audio content, we convert the auditory data into images and then apply our methods.

6 Evaluation and Discussion

6.1 Experimental Setup and Performance Metrics

Environment: To assess the efficiency of our methodology, we implemented the proposed algorithms using MATLAB R2016a, a versatile numerical computing environment known for its broad application in scientific and engineering domains. The computational simulations were performed on a Windows OS, utilizing a desktop system with an Intel Core i7-6500U CPU (2.50 GHz) and 8GB of RAM.

Dataset: Our methods are based on compression to reduce the computational cost. To measure the impact of compression, We conducted our tests using Derf-HD video sequences [28], which is provided by The Xiph.Org Foundation, a non-profit focused on promoting open and free multimedia protocols. The dataset includes video sequences in uncompressed YUV4MPEG format, compatible with tools like Theora. This dataset is widely utilized in video compression research, encoding performance tests, and quality evaluation studies.

Videos can be classified based on motion intensity [29, 30]. These classes are defined to indicate the perceived amount of motion or activity in the video. It quantifies how much motion occurs within a specific frame or over a sequence of frames. This metric is crucial in various applications, including video surveillance, video quality assessment, and video compression. Video classes are as follows:

- High Motion Intensity Videos: This category encompasses videos characterized by rapid action, dynamic scenes, and events involving substantial movement within frames.
- Medium Motion Intensity Videos: Videos falling under this category exhibit a moderate level of motion, striking a balance between dynamic scenes and static elements.
- Low Motion Intensity Videos: Videos categorized as having low motion intensity feature relatively static scenes, minimal movement, or even still images.

By testing the proposed approach across videos of varying motion intensity, *we aimed to assess its robustness and performance in different scenarios.* This diverse set of video types allowed us to analyze the proposed approach's effectiveness across a spectrum of motion intensity levels. The subsequent sections present performance metrics and offer a comprehensive analysis of the obtained results.

Performance: Our approach utilizes DWT and bit plane slicing to extract compact feature details about the multimedia resource. To evaluate the quality of the reconstructed image resulting from our approach and to discern the most suitable image plane or sub-band for hash function calculation, we employ the Peak Signal-to-Noise Ratio (PSNR) as a quantitative indicator. PSNR is a widely used measure that assesses the fidelity of an image reconstruction by comparing it to a reference image. It measures the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. A higher PSNR value indicates better reconstruction quality, as it signifies a smaller error or noise in the reconstructed or compressed image. PSNR is commonly expressed in decibels (dB) and is calculated using the mean squared error (MSE) between the original and reconstructed images. PSNR is defined as follows:

$$\text{PSNR} = 10\log[(R^2/\text{MSE})]$$

$$\text{MSE} = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{Y=0}^{N-1}(F(x,y) - F'(x,y))^2$$

where R is the maximum fluctuation in the input image data type (e.g., 255 for 8-bit images). MSE is the mean square error and $F(x, y)$ refers to the pixel of the original image, and $F'(x, y)$ refers to the pixel of the reconstructed image. A higher PSNR value indicates a smaller MSE and, consequently, a smaller error, leading to better image quality. PSNR is widely used in image and video compression to assess the fidelity of the reconstructed content.

The subsequent subsections detail the results obtained by applying these metrics to the restored images across different motion intensity levels.

6.2 Results and Analysis

Our methodology is based on bit plane slicing or DWT to extract multimedia features. It is imperative to ascertain the preservation of all key attributes of multimedia resources with minimal loss. In the bit plane slicing approach, we select plane_ 7 for hash computation, as it encapsulates the most significant image information among the available bit planes.

As discussed in Sect. 2.2.1, the intensity $I(x, y)$ of a pixel at position (x, y) is the cumulative sum of contributions from all bit planes, where each bit plane $b_k(x, y)$ is weighted exponentially by 2^k. This means that the higher the value of k, the significant contribution of that bit plane to the pixel's intensity. For example, When $k = 7$, corresponding to the MSB, the weight $2^7 = 127$ is the highest among all bit planes. So the MSB dominates the cumulative sum, contributing substantially to the pixel's overall intensity and having the greatest influence on the image's visual quality.

To validate our hypothesis, we compute PSNR and MSE for every image plane to measure the impact of compression

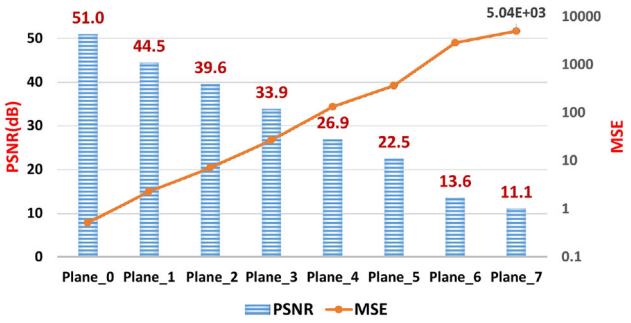

Fig. 4 Impact on PSNR and MSE when removing plane #n of image

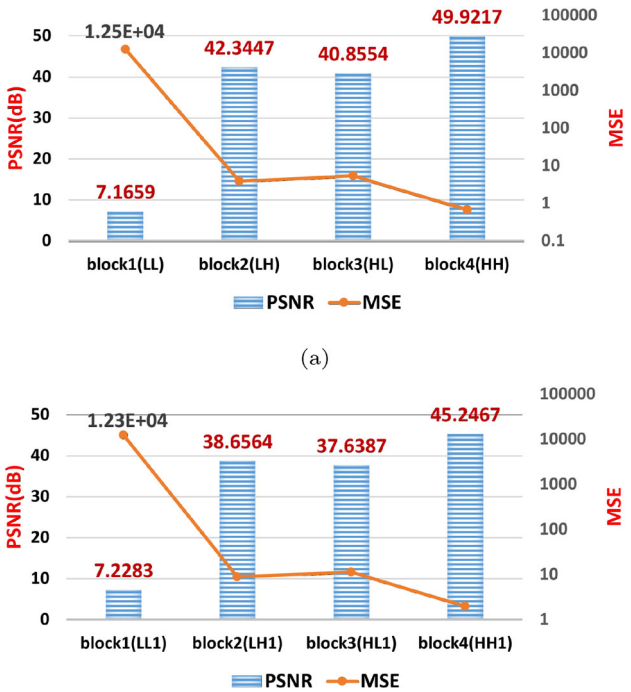

Fig. 5 **a** DWT1, **b** DWT2

of bit plane slicing compression. Figure 4 illustrates PSNR and MSE values in the context of removing plane_*n*, where *n* corresponds to the specific plane number within the bit plane slicing technique.

As shown in Fig. 4, PSNR of plane_ 7 indicates a diminished quality or degraded representation of the image or video. Removing plane_ 7 leads to noticeable artifacts and degradation in the reconstructed image. These results coincide with our assumption that plane_ 7 encapsulates the most significant image information among the available bit planes.

Summary: plane_ 7 preserves the image's critical features, making it ideal for compression and security applications. It maintains key image content even after compression, reduces redundancy, and ensures resilience against distortions.

Similarly, for the discrete Wavelet transform approach, we calculate PSNR and MSE for all sub-bands, including LL, LH, HL, and HH. Figure 5 shows the PSNR and MSE values in the context of removing one of the sub-bands (LL, LH, HL, HH) of the first and second levels of DWT decomposition. This analysis aims to determine which sub-band contains the most information.

As depicted in Fig. 5a, b, it is evident that the first sub-band within the first- or second-level decomposition of DWT exhibits a reduced PSNR value and an elevated MSE value. Consequently, we select the first sub-band as the representative multimedia component for hash computation, as it encompasses the most substantial image information among the available sub-bands.

As presented in Sect. 6.1, videos can be categorized according to motion intensity. To validate our approach, we assessed our schemes across all video classes. Table 2 demonstrates the maximum, average, and minimum values of PSNR for our approaches across all video classes. Moreover, Fig. 6 shows the average values of PSNR across all video classes, where Fig. 6a–c shows PSNR values for low motion, medium motion, and high motion, respectively. It is evident that the first- and second-level decompositions of DWT outperform bit plane slicing significantly in terms of image and video quality.

Our schemes are proposed to speed up the hash calculation. Figure 7 provides an overview of the time expended by each scheme across various video types, with Fig. 7a–c presenting the TC in milliseconds for low motion, medium motion, and high motion categories, respectively. As noticed, the second-level decompositions of DWT outperform bit plane slicing significantly and first-level DWT. Conversely, bit plane slicing surpasses the first-level decompositions of the DWT in terms of computational efficiency.

6.3 Case Study

As elaborated upon in Sect. 5, our methodology can be applied to a spectrum of multimedia resources. We have implemented our methodologies across various multimedia formats, encompassing video, audio, and text files. To quantify the reduction in processing time, we conducted a comparative analysis against the original securing scheme devoid of compression. Figure 8 illustrates the processing time in milliseconds, with Fig. 8a–c delineating the processing time for video, audio, and text files, respectively. Evidently, our proposed approach has achieved a substantial reduction in processing time, as demonstrated in Fig. 8.

This arises from the necessity to conduct a comprehensive analysis of the entire image, which entails the processing of all pixel data, thereby incurring significant computational demands and TC. For instance, in the case of the first-level DWT, a hash computation is performed specifically on the

Table 2 PSNR comparative analysis of the proposed approaches across different video classes

Video class	Approaches Video name	B-plane slicing			DWT1			DWT2		
		Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
Low motion	KristenAndSara	11.97	12.65	12.3	6.14	6.56	6.3	6.23	6.65	6.4
	Johnny	11.25	11.45	11.4	5.39	5.56	5.5	5.45	5.61	5.5
	FourPeople	10.70	10.90	10.8	6.96	7.06	7.0	6.23	6.65	7.1
Medium motion	Mobile	8.78	9.34	9.1	4.62	5.44	5.1	4.54	5.76	4.9
	Giraffes	8.70	9.62	9.2	4.48	5.26	4.9	4.54	5.38	4.9
	Elephants	7.88	9.07	8.4	5.14	5.85	5.5	5.20	5.92	5.6
High motion	Football	11.65	12.69	12.0	5.42	7.33	7.1	7.07	7.52	7.3
	Stefan	8.29	8.37	8.3	5.42	5.55	5.5	5.65	5.78	5.7

*Video parameters: Frame Rate (FR), Resolution (R), Time T (seconds) KristenAndSara: FR:60, R:1280 × 720, :10; Johnny: FR:60, R:1280 × 720, T:10 FourPeople: FR:60, R:1280 × 720, :5; Mobile: FR:30, R:386 × 288, T:5 Giraffes: FR:30,:1280 × 720, T:39; Elephants: FR:30, R:1280 × 720, T:35 Football: FR:30, R:710 × 486, T:6; Stefan: FR:30, R:320 × 240, T:5

LL sub-band, yielding a reduction in size by 25% in comparison with the complete image. This reduction effectively expedites the hash calculation process. Notably, the hashing procedure for the second-level DWT exhibits enhanced speed, owing to its sub-band being one-fourth the size of the first DWT level, further enhancing computational efficiency. On the other hand, both bit plane slicing and DWT compression methodologies are designed to retain all primary multimedia features, ensuring the preservation of MC integrity.

As noticed in Fig. 8, the bit plane slicing, first-level, and second-level decompositions of the DWT exhibit notable improvements over Hash Frame [16]. In the case of video, as illustrated in Fig. 8a, bit plane slicing reduces processing time to 13.5%, while first-level DWT and second-level DWT decrease it to 30% and 8.4%, respectively. When examining audio data, as shown in Fig. 8c, bit plane slicing reduces processing time to 18.7%, first-level DWT to 31.9%, and second-level DWT to 10.6%. In the context of textual data, as indicated in Fig. 8b, bit plane slicing reduces processing time to 11%, first-level DWT reduces processing time to 25.8%, and second-level DWT reduces processing time to 7%.

Overall, our proposed techniques significantly reduce processing time across video, audio, and text file scenarios. Specifically, bit plane slicing, first-level DWT, and second-level DWT exhibit respective TC reductions to 14.4%, 29%, and 8.6%. The selection between bit plane slicing and DWT is subject to user requirements. These approaches entail a trade-off between time and PSNR. If prioritizing PSNR, DWT may be preferred, while bit plane slicing could be chosen for faster processing times. These approaches involve a trade-off between time and PSNR.

To investigate resource consumption, we evaluate the utilized resources by our approaches for a 360-frame video across different resolutions, chroma subsampling schemes, and formats, including 4:2:2 NTSC and 4SIF. Table 3 summarizes the comparative analysis in terms of memory allocation (M) and processing time (T). The results coincide with our evaluation, where bit plane slicing offers a strong trade-off between memory usage and processing time, ensuring practicality for secure multimedia processing, and DWT2 is the most resource-efficient method, making it ideal for constrained platforms.

Table 3 Comparison of resource consumption across different approaches

Approach	Video type			
	NTSC		4SIF	
	M	T	M	T
Hash frame [16]	5244	17	3552	16
Bit plane slicing(P#7)	2092	10	2108	9
DWT1	3248	14	2856	12
DWT2	180	8	160	6

*T: processing time in m-seconds. M: memory allocation in KB. NTSC-720 × 480 pixels, 4SIF-704 × 480 pixels

Most of the schemes in related works apply hashing to all pixels of the frame, resulting in a fixed Time Complexity (TC) per frame. Consequently, for multiple frames (n), the time complexity scales linearly with the number of frames. Thus, frame processing time serves as a primary reference for comparing related approaches. While some alternative methods have attempted to reduce processing time, they often compromise security. In contrast, our approach ensures data integrity without significant trade-offs, maintaining a reasonable TC compared to these alternatives.

6.4 Security Analysis

This section presents our threat model. Subsequently, we demonstrate the robustness of our proposed approaches against integrity attacks, highlighting their potential as a

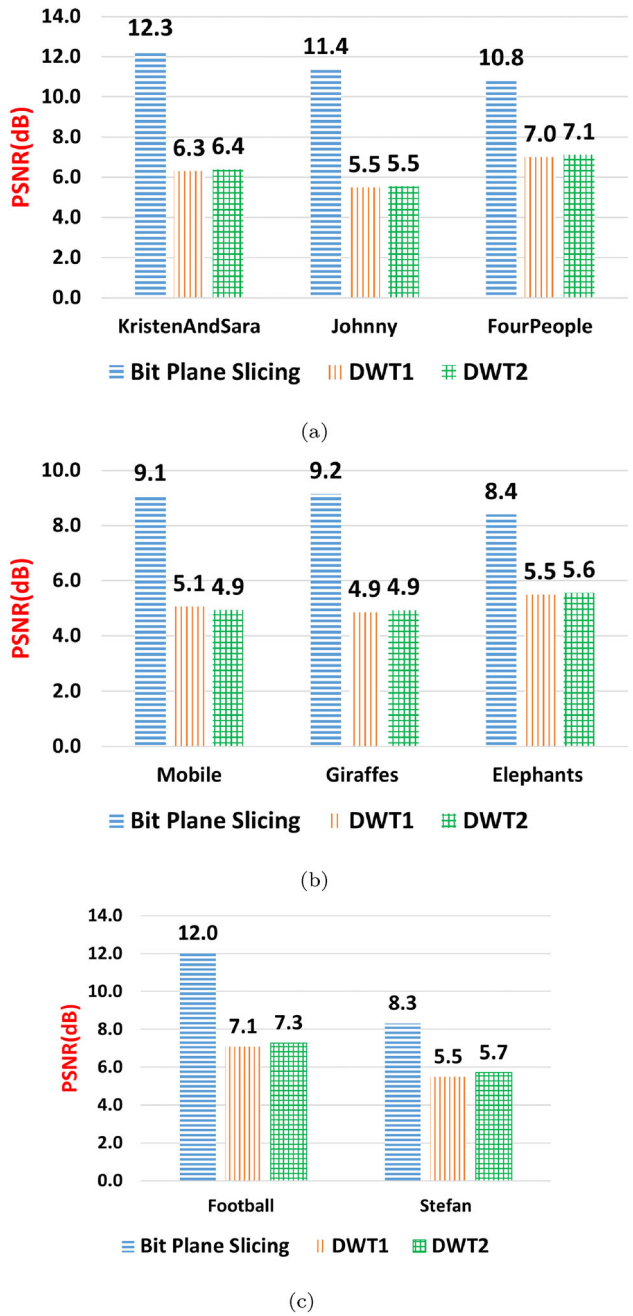

Fig. 6 **a** Low motion, **b** medium motion, **c** high motion

Fig. 7 **a** Low motion, **b** medium motion, **c** high motion

promising solution for high-performance secure multimedia platforms.

6.4.1 Threat Model

Our proposed method minimizes resource consumption while preserving key image content post-compression and ensuring robustness against integrity attacks. Our threat model focuses on securing multimedia content against *integrity violations*, addressing a range of tampering threats. These include unauthorized modifications, such as altering or injecting malicious data into multimedia files, content forgery, where attackers create falsified versions of original data, and frame-level manipulations, which compromise the authenticity of individual video or image frames.

6.4.2 Threats and Countermeasures

- Unauthorized Tampering Attack:

In this scenario, attackers modify the multimedia data after the hash has been computed and securely stored on the blockchain.

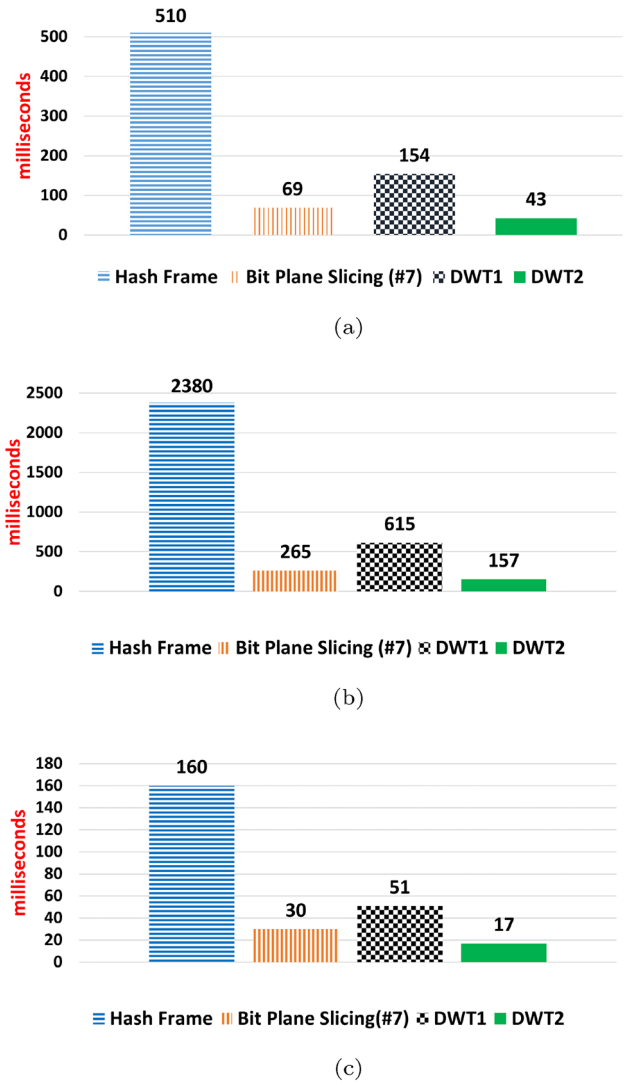

Fig. 8 **a** Video, **b** text, **c** audio

||Countermeasure: This creates an inconsistency between the hash value stored on the blockchain and the current state of the multimedia data. This inconsistency can be detected during integrity verification, as the recalculated hash of the modified data will no longer match the stored hash on the blockchain.

- Man-in-the-Middle Attack:

In this scenario, attackers intercept and modify multimedia data or its hash during transmission to the BC.

||Countermeasure: This leads to a discrepancy between the hash values stored on the blockchain and the current state of the multimedia data. As attackers cannot alter the immutable hash values on the blockchain, the modified hash fails to align with the previous hash chain. This inconsistency can be effectively detected during integrity verification.

- Replay Attack

Table 4 Hamming distance across tampering types for proposed approaches

Tampering type	Hamming distance		
	B-plane slicing	DWT1	DWT2
Face swap	0.9531	0.9063	0.9063
Frame insertion	0.9531	0.9844	0.9219
Frame deletion	0.9375	0.9375	0.9063
Clone	0.9219	0.9688	0.9688
Inpaint	0.9531	0.9219	0.9063
Splice	0.9531	0.9531	0.9844

In this scenario, attackers use valid hashes from a compromised blockchain to impersonate legitimate multimedia files, bypassing the integrity check.

||Countermeasure: as discussed and shown in Fig. 2, T_i: the timestamp of frame F_i is added to the block to establish chronological order and ensure integrity.

6.4.3 Evaluation

Tampering attacks: We simulate varied and realistic attack scenarios including authentic multimedia, deepfake, inter-frame forged, and intra-frame forged multimedia data.
Dataset: Our evaluation utilizes two key datasets:

- *Celeb-DF Dataset* [31]: contains both authentic and deepfake videos and serves as an excellent basis for studying multimedia integrity threats in video content.
- *Large-Scale Tampered Video Dataset* [32]: includes inter-frame forged data (e.g., frame insertion and deletion) and intra-frame forged data (e.g., cloning, splicing, inpainting). It is particularly suited for assessing video forgery detection techniques across a diverse range of manipulation methods.

Tampering Detection: To detect tampering, a new hash is computed for the altered multimedia data, and its validity is checked by comparing it to the BC-stored hash. The Hamming distance metric is employed to quantify the differences between the original and tampered multimedia hashes. The Hamming distance formula is as follows:

$$D_H(h_1, h_2) = \sum_{i=1}^{n} |h_1(i) - h_2(i)|$$

where $h_1(i)$ and $h_2(i)$ represent the bits of the two hash values. The value of D_H will range from 0 (identical hashes) to n (completely different hashes), where n is the length of the hash strings.

Table 4 presents the Hamming Distance values corresponding to various tampering types, including face swapping, frame insertion, cloning, and splicing. In all scenarios, the Hamming distance values are consistently greater than zero, signifying detectable differences between the original and tampered multimedia data. These results demonstrate the validity of the proposed approaches in securing multimedia.

7 Conclusion

As multimedia content continues to grow rapidly across various domains, preserving integrity has become a significant challenge. In this context, we propose a novel scheme that provides an efficient and effective solution for preserving multimedia data integrity. The proposed method integrates blockchain technology, bit plane slicing, and multilevel DWT decomposition. This combination optimizes hash block calculation for multimedia content, streamlining the process while maintaining a decentralized and secure architecture.

Evaluation results demonstrate a significant reduction in computational time, addressing a critical aspect of multimedia processing. It should be highlighted that our approach achieves these improvements without compromising multimedia content integrity. Furthermore, our approach avoids significant limitations and maintains a reasonable time complexity compared to alternative methods, highlighting its practical applicability. Future work will explore further implementations to enhance and extend the proposed approach.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

References

1. Spajic, D.J.: piracy-statistics (2023). https://dataprot.net/statistics/piracy-statistics/. Online; Accessed 3 Jan 2024.
2. Furht, B.; Kirovski, D.: Multimedia Security Handbook. CRC Press, Cambridge (2004)
3. Abou El Houda, Z.; Hafid, A.S.; Khoukhi, L.: Cochain-SC: an intra- and inter-domain DDOS mitigation scheme based on blockchain using SDN and smart contract. IEEE Access **7**, 98893–98907 (2019)
4. Abou El Houda, Z.; Hafid, A.; Khoukhi, L.: Blockchain meets AMI: towards secure advanced metering infrastructures. In: ICC 2020-2020 IEEE International Conference on Communications (ICC), pp. 1–6. IEEE (2020).
5. Sarmah, S.S.: Understanding blockchain technology. Computer Science and Engineering **8**(2), 23–29 (2018)
6. Edwards, T.: Discrete wavelet transforms: theory and implementation. Universidad de **1991**, 28–35 (1991)
7. Parveen, N.R.S.; Sathik, D.M.M.: Feature extraction by bit plane slicing technique. Int. J. Comput. Commun. Inf. Syst. **1**.
8. DeepMedia, deepfake-statistics (2023). https://www.edsmart.org/deepfake-statistics/. Online; Accessed 3 Jan 2024.
9. Sowmya, K.; Chennamma, H.: Video authentication using watermark and digital signature-a study. In: Proceedings of the 1st International Conference on Computational Intelligence and Informatics: ICCII 2016, pp. 53–64. Springer (2017).
10. Kwon, O.-J.; Choi, S.; Lee, B.: A watermark-based scheme for authenticating jpeg image integrity. IEEE Access **6**, 46194–46205 (2018)
11. Ghai, A.; Kumar, P.; Gupta, S.: A deep-learning-based image forgery detection framework for controlling the spread of misinformation. Information Technology & People (2021).
12. El-Shafai, W.; Almomani, I.M.; Alkhayer, A.: Optical bit-plane-based 3D-JST cryptography algorithm with cascaded 2D-FRFT encryption for efficient and secure HEVC communication. IEEE Access **9**, 35004–35026 (2021)
13. Qureshi, A.; Megías Jiménez, D.: Blockchain-based multimedia content protection: review and open challenges. Appl. Sci. **11**(1), 1 (2020)
14. Maetouq, A.; Daud, S.M.; Ahmad, N.A.; Maarop, N.; Sjarif, N.N.A.; Abas, H.: Comparison of hash function algorithms against attacks: a review. Int. J. Adv. Comput. Sci. Appl. **9**, 8 (2018)
15. Westerlund, M.: The emergence of deepfake technology: a review. Technol. Innov. Manag. Rev. **9**, 11 (2019)
16. Yatskiv, V.; Yatskiv, N.; Bandrivskyi, O.: Proof of video integrity based on blockchain. In: 2019 9th International Conference on Advanced Computer Information Technologies (ACIT), pp. 431–434. IEEE (2019).
17. Gallo, P.; Pongnumkul, S.; Nguyen, U.Q.: Blocksee: blockchain for iot video surveillance in smart cities. In: 2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), pp. 1–6. IEEE (2018).
18. Ghimire, S.; Choi, J.Y.; Lee, B.: Using blockchain for improved video integrity verification. IEEE Trans. Multimedia **22**(1), 108–121 (2019)
19. Mercan, S.; Cebe, M.; Aygun, R.S.; Akkaya, K.; Toussaint, E.; Danko, D.: Blockchain-based video forensics and integrity verification framework for wireless internet-of-things devices. Security and Privacy **4**(2), e143 (2021)
20. Danko, D.; Mercan, S.; Cebe, M.; Akkaya, K.: Assuring the integrity of videos from wireless-based iot devices using blockchain. In: 2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW), pp. 48–52. IEEE (2019).
21. Lawrence, L.; Shreelekshmi, R.: Chained digital signature for the improved video integrity verification. In: Modern Management Based on Big Data II and Machine Learning and Intelligent Systems III-Proceedings of MMBD 2021 and MLIS 2021. China, vol. 341. IOS Press, pp. 520–526 (2021)

22. Ghimire, S.; Lee, B.: A data integrity verification method for surveillance video system. Multimedia Tools Appl. **79**, 30163–30185 (2020)
23. Jeong, Y.; Hwang, D.; Kim, K.-H.: Blockchain-based management of video surveillance systems. In: 2019 International Conference on Information Networking (ICOIN), pp. 465–468. IEEE (2019).
24. Koptyra, K.; Ogiela, M.R.: Imagechain-application of blockchain technology for images. Sensors **21**(1), 82 (2020)
25. She, W.; Huo, L.; Tian, Z.; Zhuang, Y.; Niu, C.; Liu, W.: A double steganography model combining blockchain and interplanetary file system. Peer-to-Peer Network. Appl. **14**(5), 3029–3042 (2021)
26. Lee, W.Y.; Choi, Y.-S.: Reliable integrity preservation analysis of video contents with support of blockchain systems. Appl. Sci. **12**(20), 10280 (2022)
27. Fitwi, A.; Chen, Y.: Secure and privacy-preserving stored surveillance video sharing atop permissioned blockchain. In: 2021 International Conference on Computer Communications and Networks (ICCCN), pp. 1–8. IEEE (2021).
28. Montgomery, C.; et al., (2021) Xiph. org video test media (derf's collection), the xiph open source community, 1994. Online, https://media.xiph.org/video/derf, 3, 5.
29. Mary Idicula, S.; et al.: Enhanced video classification system using a block-based motion vector. Information **11**(11), 499 (2020)
30. Ma, Y.-F.; Zhang, H.-J.: Motion pattern-based video classification and retrieval. EURASIP J. Adv. Signal Process. **2003**, 1–10 (2003)
31. Li, Y.; Yang, X.; Sun, P.; Qi, H.; Lyu, S.: Celeb-df: a large-scale challenging dataset for deepfake forensics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3207–3216 (2020).
32. Singla, N.; Singh, J.; Nagpal, S.; Tokas, B.: Hevc based tampered video database development for forensic investigation. Multimedia Tools Appl. **82**(17), 25493–25526 (2023)