https://doi.org/10.2352/ISSN.2470-1173.2020.4.MWSF-117

A System for Mitigating the Problem of Deepfake News Videos Using Watermarking

Adnan Alattar, Ravi Sharma, and John Scriven, Digimarc Corporation, 9405 SW Gemini Drive, Beaverton, OR, USA 97008-7192

Abstract

This paper describes how watermarking technology can be used to prevent the proliferation of Deepfake news. In the proposed system, digital watermarks are embedded in the audio and video tracks of video clips of trusted news agencies at the time the videos are captured or before they are distributed. The watermarks are detected at the social media network's portals, nodes, and back ends. The embedded watermark imparts a unique identifier to the video, that links it to a blockchain. The watermarks also allow video source tracking, integrity verification, and alteration localization. The watermark detectors can be standalone software applications, or they can be integrated with other applications. They are used to perform three main tasks: (1) they alert the internet user when he watches an inauthentic news video, so that he may discard it, (2) they prevent a Deepfake news video from propagating through the network (3) they perform forensic analysis to help track and remove Deepfake news video postings. The paper includes Proof-of-Concept simulation results.

1. Introduction

Concerns about the authenticity of news (text, audio, and video) distributed over the internet have reached an all-time high. In the past, people trusted news that came from reputable newspapers and trustworthy Radio/TV stations, but nowadays they can't always trust news distributed on the Internet. The Internet has enabled a non-linear media distribution model that does not guarantee the authenticity of the news. Internet users can digitally alter news of authentic sources and re-distribute them through social media networks (e.g. YouTube, Facebook, Twitter, etc.) as if they were originals coming from legitimate sources. Usually, the alteration is done in three different ways. The first is known as face-swap, in which the original face in the video is replaced with another face. The second is known as lip-sync, in which the speaker's voice is replaced by the voice of an impersonator. The third type is known as puppet-master, in which the person in the video is animated to do a desired action.

The news authenticity problem is exacerbated with the advent of deep learning technology. New powerful video creation software tools have recently been developed using deep learning and made available on the internet for free. These tools are based on the Generative Adversarial Networks (GAN) [1]. These tools made the talents and the expensive software and hardware, usually used in the movie industry, no longer required for video content altering. They run on an ordinary personal computer (PC), and their use is straightforward. A novice user can use them to quickly alter the looks, the speech, or the actions of the people filmed in any video and generate fake videos that look convincingly real. The generated fake videos are commonly known as Deepfakes and their pervasiveness on the internet has doubled in the nine months period from Dec 2018 to July 2019 [2].

This rapid increase in number of Deepfakes is alarming and their use could be detrimental to society. They have been used extensively for pornography and, to a much lesser extent, for cyberbullying celebrities, mocking renowned politicians, and robbing financial institutions. Moreover, there is a growing concern that their harmful use could substantially increase. They could be used to spread fake news to influence elections and undermine democracy. They could be used to launch misinformation attacks that could threaten national security. Their malicious use could ultimately lead the public to lose their confidence in what is real. Therefore, there is a multi-faceted interest in detecting and preventing the proliferation of ill-intentioned and malicious Deepfakes, especially video clips of fake news.

Current laws and policies are not adequate to contain the problem of Deepfakes [3] [4]. The existing information privacy laws, the defamation laws and the Digital Millennium Act (DMCA) have recently proved to be insufficient in dealing with the Deepfakes problem. Therefore, the US Congress and many states are introducing new legislation and policies to criminalize malicious Deepfakes. Also, governmental agencies are defining procedures for reporting misuse of Deepfakes, and they are also making these procedures obvious and accessible. Moreover, non-profit organizations are running national campaigns to educate the public on how to deal with the danger of Deepfakes. These legislative actions and educational efforts will help fight Deepfakes, but they are not adequate by themselves. Therefore, it is imperative to develop an advanced technical solution that would detect and prevent Deepfakes from penetrating the social media networks.

2. Background

Researchers have been investigating developing automatic detectors that detect Deepfakes from the tell-tale signs of alteration. They designed algorithms based on unusual image artifacts [5] and inconsistent image statistics, geometry, or semantics. Koopman and et al. [6] investigated the difference between the Photo Response Non-Uniformity (PRNU) of authentic videos and that of Deepfake videos. Yang and et al. [7] used a Support Vector Machines classifier (SVM) to exploit the inconsistency in head poses. Li and Lyu [8] used convolutional neural networks (CNNs) to exploit face-warping artifacts. Agarwal and Farid [9] used an SVM classifier to exploit inconsistency in facial expression and movement. Li and et al. [10] used a Long-term Recurrent Convolutional Networks (LRCN) to exploite blinking patterns. Guera and Delp used a Recurrent Neural Network (RNN) to exploit frame-level features extracted using a CNN [11]. These techniques showed very good success, but as Deepfake generation algorithms improve, alteration tell-tale signs will gradually disappear, and the developed detection techniques will become less effective.

Researchers are also investigating active techniques [12], that could be used to protect images of celebrities, famous politicians, or ordinary people from being used as targets for Deepfakes. They are

proposing embedding invisible noise in these images as the user posts them to the social media network. This noise is carefully designed to mislead the training process of the GAN network. The noise would cause the training algorithm to misregister the facial features (i.e. eyes, nose, and mouth) and use other image parts instead. This would force the Deepfake algorithm to generate Deepfake images of inferior quality that could be easily detected and hence discarded by the viewer. This research is still in its infancy stage. Therefore, its effectiveness can't yet be judged.

More effort to support, facilitate, and accelerate the development of advance algorithms for detecting Deepfakes is currently underway. Databases that contain many thousands of Deepfake videos have recently been created by media giants (e.g. Google, Facebook) and made available to researchers [13]. These databases will allow researchers to easily train, test, and benchmark the Deepfake detection algorithms being developed. Moreover, contests and challenges to incentivize researchers and accelerate development of Deepfake detection algorithms have started under the sponsorships of Facebook/Microsoft (DFDC) [14] and DARPA (SemaFor) [15]. These contests are also developing procedures for benchmarking Deepfake algorithms.

Several startup companies such as Truepic [16], Serelay [17], and Prover [18], are providing services to certify the veracity of the media (image or video) upon its capture. Each of these companies provide its subscribers with a special application for capturing the media. This application is designed to automatically send the captured media along with its metadata (capture time, location, and device) to the company's server immediately after capturing. The company, in turn, verifies the integrity of the received media, stores the media or its fingerprint in a blockchain network, and publishes a QR-code or a link to reference the media. This link can be used later by any party interested in checking the authenticity of that media.

This paper proposes a system for combating Deepfake news videos. It targets detecting fake news clips generated from existing authentic video clips using the processes of face swapping and voice impersonation. The system is based on Digimarc's audio and image watermarking and blockchain technologies which make the system simple, reliable, and efficient.

The paper is organized in eight sections including the introduction and the background (sections (1) and (2), respectively). Section (3) describes the overall system and its integration with social media networks. Section (4) gives an overview of the audio and image watermarks used by the system. Section (5) describes the video hashing process and the blockchain construction. Section (6) discusses watermark copy attack and methods to mitigate it. Section (7) provides proof-of-concept simulation results and discussions. The last section (8) includes some conclusions.

3. System Description

The objective of this research is to establish a system based on watermarking technology that detects Deepfake news generated from existing authentic video clips via face-swap and lip-sync.

The proposed system requires that trusted news entities embed unique digital watermarks in their video during video capturing or before video distribution. They can automatically serialize and embed the watermark at the time of streaming or downloading of the video by a recipient. The serialization has the extra benefit of enabling tracking distribution of copies or derivative works of content. The embedded watermarks are used to check that news videos, coming from random internet sources or appearing at the portals of social media networks, have not been altered. The watermarks are also used at the back end of the social media networks to perform forensic analysis of videos suspected to be Deepfakes. The news entity also records the history and provenance of the news in a blockchain to provide helpful inputs to the forensics process. The watermark establishes a permanent link between the video and its blockchain. Embedding a serialized watermark payload in each copy or derivative work enables each instance of content distribution to be appended in a block of the blockchain. The system enables authorized users to access the original un-watermarked video, create an authorized derivative work, and then watermark and append it to the blockchain.

The watermarks are embedded in the audio and video tracks at fine granularity and specificity to allow reliable detection at an extremely low false positive rate. The embedded marks are robust to common video manipulations, such as D/A and A/D conversion, filtering and compression, but they are sensitive to changes that target the video integrity. They impart unique identities to the host video linking the video to its source and provenance in a blockchain network. The watermarks allow comparing the video with its original copy at the source to establish its authenticity. The video and audio watermarks are tightly coupled, and they cross reference each other to verify the integrity of the video. This coupling allows these watermarks to verify that an audio segment has not been replaced as is often done in making Deepfakes via voice impersonation. The frame watermark can localize frame changes resulting from face swapping, which is commonly used for making Deepfakes.

Once the video is watermarked, the news entity can provide its users with a software application to view and authenticate videos coming from random sources. While a user watches the video, the software automatically attempts to detect and decode the watermarks if they exist. The software performs an integrity and consistency check between the video and audio watermarks and determines authenticity according to a specific integrity criterion. A green light is used to indicate authentic videos, a red light is used to indicate fake videos that do not satisfy the integrity criterion, and an amber light is used to indicate unwatermarked videos. The user may also interrogate the software to determine and localize the changes introduced in the video. In this case, the software performs a forensic analysis and compares the video-under-test with the original video stored in a publicly accessible distributed blockchain network to determine the altered or missing content. This allows the viewers to identify and discard any Deepfake news video clips and distrust its source.

The detection software can also be integrated with popular media players to enable them to check the watermarks and report authenticity to the viewer as the dedicated player software does. The watermark detector software can also be used at the portals of social media networks to prevent doctored videos from spreading widely. It can also be integrated with the back end of content hosting and sharing networks, including social media networks, to perform forensic analysis of the suspected video to determine whether the video should be taken down or not.

4. Digmarc's Watermarks

The system described in this paper employs two kinds of robust watermarking technologies to identify and authenticate the video.

The first is an audio watermark, and the other one is an image watermark, available from Digimarc Corporation. Both marks are used in a tightly coupled way to provide extended payload capacity, unique content identification, and an adequate level of security. Only authorized entities (e.g., news agencies) have access to the watermark embedder. Also, a secure watermark reader employs detection protocol known only to an authorized party, such as a content owner and his delegates.

4.1. Audio Watermark

The audio watermarking technology is used to protect the audio component from alteration or separation from the original video [19]. For this application, an audio watermark encoder embeds an imperceptible spread spectrum signal (e.g., in the 120 Hz to 8 kHz range) into the magnitude of the frequency coefficients of each channel of the host audio. The host audio is typically sampled at 44.1 kHz or 48 kHz. The encoder generates watermarked audio by first processing the host audio as frames of 2048 consecutive samples at 16 kHz sampling rate. A spread spectrum watermark is embedded in the frequency representation of each of these frames. A different payload is embedded in every one second of audio using an extensible payload protocol that enables the deployment of different watermark versions. The fine granularity of the watermark allows the system to detect fine alteration of the audio track.

For the proof of concept, the watermark signal is a concatenation of 3 bits for version control, 24 bits for CRC, and 24 bits for variable payload. Furthermore, the CRC and payload bits are encoded using convolutional and repetition encoding to protect them from channel errors. Repetition only is used to protect the version bits from error. A unique pseudo-random sequence (PN) is also used to scatter the resulting sequence of bits to make them look like noise. The scattering process makes the watermark imperceptible and provides additional protection against erasure error. The PN sequence also serves as a security key that can be chosen uniquely per user or per application to provide serialization. Finally, the resulting bit-sequence is scaled and shaped according to a psychoacoustic model to provide masking during audio playback. The result is a watermark signal of 2048 samples ready for embedding in the host audio channel.

To embed the resulting watermark signal in the host audio, the host audio frame is transformed to the frequency domain using the Fourier transform and the watermark is added to the magnitudes of the Fourier coefficients. The sign of the watermark is reversed in every other frame. The bit reversal allows the detector to reduce the effect of the host audio by subtracting every two consecutive frames from each other before decoding. This subtraction cancels the host signal and reinforces the watermark signal which enhances the signal to noise ratio. Finally, the embedded frequency coefficients (magnitudes and phases) are transformed to the time domain using the inverse Fourier transform to generate the embedded audio. Several variants are possible. For example, the watermark may be adapted based on frequency and time domain perceptual modeling, and then inserted into the audio signal in the time or frequency domain [20]. Real time, low latency encoding may be employed to enable transactional watermarking at the time of transmission of the video [21].

Detection of the audio watermark is performed at 16 kHz sampling rate using one second of audio. Frame accumulation with sign reversal every other frame is first performed to boost the signal-to-noise ratio. Synchronization is achieved by correlating the audio with fractional shifts of a watermark frame. The accumulated signal is then transformed to the frequency domain using the Fourier transform. The Fourier magnitudes are calculated and correlated with the PN spreading sequence to obtain the encoded payload sequence. The version bits are first decoded from the encoded payload sequence. Then the Viterbi convolution decoding is performed to correct for any errors and the CRC bits are recalculated to verify the presence of the watermark. Finally, the payload bits are decoded.

The audio watermark can be detected in as little as one second of audio, but longer duration is needed for increased reliability. The watermark is robust to noise, compression, D/A and A/D conversion, and the broadcast/streaming environments. It can also be detected in the presence of linear time scaling and pitch invariant time scaling.

4.2. Video Watermark

The image watermarking technology is used to protect the video frames from alteration. It is embedded into the frames of the video clips in either the uncompressed or compressed domain (e.g. MPEG4) [22] [23]. The watermark consists of a synchronization signal and a payload signal. The synchronization signal is embedded in the frequency domain and the payload signal is embedded in the spatial domain. The two signals are added together according to a predetermined ratio to form a 128x128 tile. This tile is embedded into each video frame by simple addition and tiled to cover the entire frame. Before addition, the strength of the tile is adjusted according to the local characteristics of the frame and the overall desired robustness level. Also, the tile could be up-sampled to a different size, to make it better suited to the deployment environment. Different frames carry different payloads to allow the detection of frame insertion, deletion, and shuffling.

The synchronization signal is a constellation of frequency peaks of the same magnitudes and random phases. These frequency peaks form a pattern in the frequency domain and are used to guide the detector in reading the watermark. The watermark reader uses this frequency pattern to reverse the affine transformation that results from video manipulations such as rotation, scaling and cropping. The payload protocol is extensible and has similar structure to that of the audio watermark. For the proof of concept, the payload signal consists of 75 bits composed of 4 bits for version control, 24 bits for CRC, and 47 bits for the variable payload. The version bits are protected from error using convolutional encoding and repetition while the CRC and payload bits are protected against channel error using only convolutional encoding. Each bit of the resulting sequence is also spread and scattered 16 times within a 128x128 tile using a unique PN sequence. As in the audio watermark, these PN sequences can be chosen uniquely per user or per application to provide serialization.

The watermark can be independently detected and read from each 128x128 block in each frame of the video. First a non-linear filter is used to separate the watermark from the host frame. Then, the presence of the synchronization signal is detected in the frequency domain using a match filter or least square fitting. Then the block's scale, rotation angle, and translation parameters are estimated. These parameters are used to properly align the frame block for reading the payload signal. The payload bits are extracted from the aligned block and the scattering and spreading process is reversed. The version bits are then decoded, and the repeated bits are accumulated to enhance the signal to noise ratio. Then the Viterbi decoding is applied to obtain the variable and CRC bits. The CRC bits are recalculated and compared to the decoded CRC bits.

Correct CRC bits indicate successful reading of valid variable payload bits.

5. Blockchains

The system uses blockchains to store all the information needed for performing forensic analysis on a suspected news video clip. This information includes copies of all published editions of a video clip and their relevant metadata [24] [25]. Metadata includes information that is created by the capture hardware or editing software (e.g. file name, file type, GPS coordinate, camera settings, time stamp, duration, ownership, etc.). It also includes human generated information that describe the video (e.g. keywords, tags, and comments). It also includes information generated automatically by speech and image recognition software (e.g. video transcripts, shots' boundaries and descriptions, Scale-Invariant Feature Transform (SIFT) key points [26], video and audio fingerprints, cryptographic hash, etc.). Different types of blockchain systems are used for storing the videos and their metadata. The metadata can be retrieved based on the watermark embedded in the video.

A blockchain is a distributed, transparent, and publicly verifiable ledger composed of a series of immutable blocks of data records that are replicated and stored in a network of distributed computers. Each of these blocks contains one or more transaction records and a cryptographic hash. This hash is calculated from the data in the previous block including the hash of its predecessor block. These hashes make the blocks in the chain practically immutable. Any change made to an existing block would require recalculating and changing the hashes in all subsequent blocks in all the computers of the network (nodes). This recalculation is practically impossible, especially in a large network of many computers storing large number of blocks. The nodes in a blockchain network are used to record transactions in blocks, store these blocks, verify transactions, and manage the overall ledger.

The blockchain network can be decentralized or centralized. Decentralized networks allow anonymous users to participate and transact on the ledger. The Proof-of-Work (PoW)/mining mechanism is used to maintain the integrity of the ledger and prevent malicious users from corrupting the system. On the other hand, centralized networks allow only credible participants (authorities) to transact on the ledger. The identities of these participants are known, and their transactions can be audited at any time. The authentication mechanism used by these centralized networks is known as Proof-of-authority (PoA). Compared to PoW, PoA networks are more secured, less computationally intensive, more performant, and more predictable. Therefore, the centralized blockchain networks are more appropriate for use in our system than the decentralized blockchain networks, but decentralized blockchain networks may also be used.

Blockchains are inherently not suitable for storing a large amount of data such as video data. Because blockchains replicate the ledger on each of their nodes, storing video databases on them requires extremely expensive storage hardware. Moreover, most blockchains impose limits on their block size and rate of appending blocks to the network. The block rate limit protects the network from the double spending problem, and the block size limit makes the PoW mechanism effective. Bitcoin limits the block size to one Mega Byte and the block rate to one block every ten minutes. On the other hand, an Ethereum network has no limit on the block size in the blockchain, and it has an increased block rate of one block every fifteen seconds. Changing the block size and block rate is tricky, and if not done carefully, it could affect the security of the blockchain [27].

To avoid the aforementioned problems, our system does not store the video data in an ordinary blockchain. Our system stores the video data in a Distributed Storage Platform (DSP). The Inter-Planetary File System (IPFS), Swarm, Sia, Storj, or MaidSafe are popular examples of DSP. These platforms are effective peer-to-peer systems that store their data in a distributed, safe, robust, and decentralized manner without duplication. They are based on the Ethereum blockchain technology, which is used to incentivize participants to pool their resources (i.e. storage and bandwidth) and provide them to all participants of the network in the exchange of monetary compensation. A DSP, from a developer point of view, is similar to the World-Wide-Web, except that the uploads are not hosted on a specific server. Instead, chunks of the uploaded file are hashed and stored on different servers. A Distributed Hash Table (DHT) is used internally to retrieve the data chunks from these servers. A root hash, in machine and human readable format that serves as a Content Identifier Number (CID), is used externally to identify and retrieve the entire file. A DSP usually tracks changes to its files using a separate blockchain network, which allows the users of our system to retrieve the change history and provenance of the video file using the same hash.

Unlike video data, the metadata of the video is stored in a private centralized PoA-based Ethereum blockchain network in the form of transaction data or a smart contract. This network is fast, economical, and onymous. It contains a limited set of nodes; each of which is controlled exclusively by an authentic news publisher. These publishers are the only users who can transact to the network. Other users can only retrieve and view the information already written to the blockchain. The standalone watermark readers or the readers integrated within the media players or the networks' back end forensic tools are technically users of the blockchain network with only read access-rights. Each block in the blockchain is restricted to contain only one transaction and each transaction is restricted to be related to only one video. These limits on the blocks and their transactions provide a one-to-one correspondence between the Video Identification Number (VIN) and the block number in the blockchain.

The VIN is included in the video watermark and is used for any forensic analysis performed on a suspected video. After decoding the payload from the watermark, the VIN can be used as an address to retrieve the CID and the metadata of the video from the centralized blockchain network. Since the blockchain is accessible by the public, this operation can be performed by any user with the proper watermark reader. The retrieved CID can then be used to retrieve the video from the IPFS. The suspected video can be viewed and compared manually to the retrieved video to determine its authenticity. The comparison can also be done automatically using an algorithm designed for this purpose. The retrieved metadata provides additional information that helps the forensic process. History and provenance information of the video can be provided by storing the information in a smart contract [28] rather than transaction data.

A traditional centralized database could be used instead of a blockchain for storing the video and its forensic information, however, using a blockchain is preferred. Blockchains eliminate the need for an expensive database administrator, who can be trusted by all participants. They provide invaluable protection for the data by

maintaining its integrity and decentralizing its storage in a highly fault-tolerant network. They create an absolute trust in the stored data, that is necessary for facilitating collaborations and enabling partnership among business associates and competitors. They store the data in immutable, transparent, and secure blocks, and they do not allow changing the data recursively. They track changes and record history of the recorded data to provide an audit trail that enables forensic analysis. Centralized databases lack these advantages; therefore, using a centralized database instead of a blockchain should only be considered an interim step in the process of implementing the proposed system, and migration to a blockchain should be the ultimate goal.

6. Copy Attack

The "Copy Attack" allows a user to estimate and extract a watermark from one video and insert it into another [29]. Therefore, an adversary could generate a Deepfake news video based on an authentic video clip then add authenticity to it by copying a watermark from another authentic watermarked video. For puppet-master Deepfakes, the watermark needs to be copied everywhere, but for face-swap Deepfakes, only the watermark on the original face region, which was replaced, needs to be copied to the new face region. Similarly, for lip-sync Deepfakes, only the watermark from the original audio segments, that were replaced, needs to be copied to the new audio segments. Consistent watermark synchronization should be preserved when the watermark from a video frame region or an audio segment is copied.

The system needs to defeat the copy attack by employing video features that would be altered by swapping in new content [30], like a face or an audio segment. A robust hash derived from the video can be used for this purpose. A hacker can blindly copy the watermark from one area into another area of a frame, but he has no way to check whether these features have been altered by the copy operation. The hash can be stored as metadata in the blockchain or included in the payload of the watermark. Making the watermark content dependent is a convenient solution, but it is not necessary for defeating the copy attack when there is access to content features in the blockchain for authenticating the video. A content dependent watermark allows video verification when access to the blockchain is not available. Therefore, we propose to include a hash of some video features in the payload. All other metadata stored in the blockchain can be used for video verification whenever access to the network is available.

The payload of the image watermark is designed to include a robust Video Frame Hash (VFH) calculated from the locations of the most prominent and robust features in a video frame. The locations of the center of the eyes, tip of the nose, and the left and right corners of the mouth could be used with portrait images [31]. Also, the areas within the boundaries of these features could be used. The MTCNN (Multi-Task Cascaded Convolutional Networks) algorithm is used for calculating these locations [32]. The payload of the audio watermark is also designed to include a robust Audio Segment Hash (ASH) calculated from the lowest quantized Mil Frequency Cepstrum Coefficients (MFCC) of the audio frame [33]. After the watermarks are decoded, the detector software recalculates these hashes and compares them with those values extracted from the payload or retrieved from the blockchain. A no-match condition would indicate a copy attack and invalidate the watermarks, hence the news video.

7. Simulation Results and Analysis

Parts of the proposed system have been implemented as proof-of-concept for the system. These parts are described in this section.

7.1. IPFS for Storing Video Data

To simulate storing the video data in a distributed storage platform, we used the Inter-Planetary File System [34]. IPFS was selected because it is free, popular, public, and designed specifically for storing digital assets (i.e. text, audio, and video). However, IPFS is still a prototype subject to change, and its participating nodes are volunteers. Therefore, storing the data in the IPFS carries the risk of losing the data if a node decides to stop participating in the network. Also, users will start paying a very reasonable charge for storing their video once the system is finalized. The IPFS system divides the file into blocks and stores them into a set of distributed nodes without duplication. This considerably reduces the storage requirement and its associated cost. We stored a sample video in the IPFS and obtained a Content Identifier (CID). The IPFS generated the CID from the video content itself. The IPFS calculates the CID from a Merkel-DAG tree representing the hashes of the chunks of the video. Although calculated differently, the CID is a 256-bit multi-hash similar to the popular SHA2-256. The CID can be used to reference the video and to authenticate any digital copy of it.

7.2. Rinkeby Blockchain for Storing Metadata

To simulate storing the metadata of the video in a blockchain, we used the popular Rinkeby testnet. Other networks such as the Ropsten and Kovan testnets or the Ethereum mainnet could also be used. Rinkeby as well as Kovan are based on PoA, but the Ropsten and Ethereum are based on PoW. The data can be stored in these networks as either smart contract or transaction data. For simplicity, we stored the data as transaction data in Rinkeby network. Reference [28] describes how to store the data as a smart contract in an Ethereum network. We used the MetaMask digital wallet for submitting transactions to the Rinkeby network. We used an older version (3.13.8) of MetaMask because the interface of the current version (7.7.1) does not have a field for entering the transaction data. We obtained the needed currency (Eth) for posting transactions from the Rinkeby Faucet.

The transaction data consisted of the CID hash of the video clip and a simple record of metadata needed for authentication. We first converted the transaction data from ASCII format to the required hexadecimal format. We then included the result in a transaction and submitted it to the network. The network queued the submitted transaction with transactions submitted by other participants. Then the network stored these transactions in a block and appended the block to the blockchain. The network assigned an identity number (BID) to the block and a transaction number (TN) to our transaction within the block. We concatenated the BID and TN and formed the video identification number (VIN). Then we embedded the VIN in the video watermark.

Because we added the watermark to the video after storing the file in the IPFS, the CID of the watermarked video would not match its CID in the blockchain. One solution to this problem is to store the CID in the Inter-Planetary Name System (IPNS) [34] and replace the CID in the blockchain with a pointer to the CID location. The IPNS is a system for creating and updating mutable links to IPFS contents. It stores the CID in an encrypted form using a pair of public and private keys. It uses a hash of the public key associated with the record containing the CID as a pointer to CID record. The stored CID is signed by the corresponding private key. After the

watermarked video is added to the IPFS, the CID record in the IPNS is replaced with the CID of the watermarked video using an update process to the IPNS. The IPNS can't keep both CIDs at the same time. To keep the CID of the original video, a JSON bundle that includes the CID of the original video and the CID of the watermarked video must be generated first using the IPFS Linked Data (IPLD) [35], and the CID of the bundle is stored in the IPNS instead of the CID of the original video. This method allows the original video as well as the watermarked video to be retrieved during the forensic analysis.

7.3. Watermark Payload

In the proof of concept embodiment, the payload of the audio watermark is changed every one second of audio, and it includes 24 bits of the following information:

1. 5 bits for Audio Segment Number (ASN): ASN is reset every 32 seconds of audio. It is used to detect audio segment deletion, insertions, and reordering. Gaps in the sequence of ASNs indicates missing audio segments, inserted segments do not have watermarks, and out of order ASN sequence indicates audio segments shuffling.
2. 14 bits for Audio Segment Hash (ASH) described in Section (6): ASH is used to protect against copy attack. A miss-match between the ASH calculated from an audio segment and the ASH in the watermark embedded in that audio segment indicates a copy attack.
3. 5 least significant bits of the Video Identifier Number (LVIN) described in Sections (5) and (7.2): A miss match between these bits and the corresponding bits of the VIN in the frame watermark indicates that the audio does not belong to the same video.

The payload of the image watermark is changed every frame, and it includes 47 bits of the following information:

1. 5 bits for Video Frame Number (VFN): VFN is reset every group of 32 consecutive frames. Gaps in the sequence of VFNs indicates missing frames, inserted frames do not have watermarks, and out of order VFN indicates frame shuffling.
2. 22 bits for Video Frame Hash (VFH) described in Section (6): A miss-match between the VFH calculated from the facial features of a video frame and the VFN in the watermark embedded in that frame indicates copy attack.
3. 20 bits for the Video Identifier Number (VIN) described in Sections (5) and (7.2): The VIN extracted from the image watermark alone is enough to retrieve the video and its metadata from the blockchain network for forensic purposes.

7.4. Deepfakes Generation and Face-Swap Detection

To evaluate the effect of the Deepfake algorithms on Digimarc's image watermark, we embedded ten frames of a head and shoulder video sequence (captured in house) and subjected them to Deepfake creation. We used the open source DeepFaceLab algorithm to replace the faces in these frames with the faces of a target person. We used a 120-frame video of the target person. We first used the TensorFlow-based MTCNN algorithm to detect and extract the faces in the original and target videos. The MTCNN is a three-stage algorithm that detects the bounding boxes of all faces in an image along with the locations of their five landmarks (two eye center, one nose tip, and two mouth corners). We used the Face

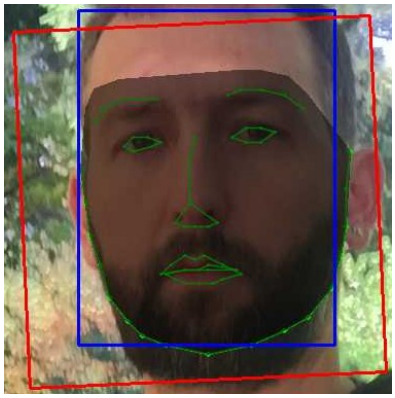


Figure 1. Face Bounding Box and its 68 Landmarks.

Figure 2. Original Frame (Left), Deep Faked Frame (Right).

Alignment Network (FAN) algorithm [36] to refine the area within the bounding box and locate 68 detailed facial landmarks as shown in Figure 1. These landmarks are later used to re-assemble the fake images. Then we used DeepFaceLab [37] to organize the extracted face images by their yaw angles. This organization aligned the faces and simplified the generation of the required Deepfake model.

We trained the DeepFaceLab on the original and target faces using an ordinary Intel Core-7 PC and applied the result to the original frames. The training process to map the target faces to the original faces was very slow (3800 iterations were performed in 26 hours). However, applying the results of the training stage to the original frames was very fast. At the end, a reasonable Deepfake was generated; an example of which is shown to right of the original frame in Figure 2. A better Deepfake could have been obtained faster using the CUDA programing language running on a modern NVIDIA GPU card, which can run iterations much faster (an order of magnitude faster than a typical PC).

We ran two experiments to create a robust VFH from the frame features to prevent the previously described copy attack. We first tried to use the estimated eyes' centers, nose tip, and mouth corners, but we found these measurements sensitive to minor image manipulations such as image blurring, sharpening, and compression.

Therefore, these features are not suitable for calculating a robust hash that can be used to prevent a copy attack. We then used the areas within the estimated boundaries of the eyes, nose and mouths. We quantized these areas with a uniform quantizer with a step size of 20 to allow a 20 square pixel error tolerance. Our preliminary results showed that these measurements are robust to ordinary image manipulations, but they are not robust to the Deepfake process. Therefore, we used them to calculate the desired 22 bits robust hash using a Python built-in hash function and a modulo operation. The Python function implements a simple multiplicative/XOR operations, and the modulo operation was used to limit the hash size to 22 bits. These bits are used as VFH and included in the image payload. They could also be stored in the metadata in the blockchain.

Finally, we used Digimarc's watermark reader on the frames of the resulted Deepfake video. The results showed that the watermark can be detected everywhere in the frames except in the face areas where the faces had been swapped. This means that the Deepfake transformation and the face swapping can be localized using image watermarking, provided that the watermark is not copied to the original frame using a copy attack. If the watermark was copied to the face area, then we would run the MTCNN algorithm on the fake images to locate the Facial area and the FAN algorithm to re-generate the 68 detailed landmarks. We would also calculate a hash of the areas of the main facial features and compare it to the hash embedded in the watermark. In this case, the comparison would fail; but it would succeed if the video was not fake. Therefore, embedding the hash of video feature is a good counter measure for the copy attack.

7.5. Deepfakes Detection and Forensics

The system performs two stages of authentication to detect fake news video clips made from watermarked video clips by audio impersonation or face replacement. The system automatically performs the first stage solely using the embedded watermark and performs the second stage only when access to the metadata in the blockchain is available. The second stage can be performed automatically or upon the user's request.

The first stage of authentication uses information embedded in the watermark. This stage does not use meta data in the blockchain. The system first looks for the watermark using a watermark reader. If both audio and video watermarks are not found, the system reports that authenticity of the video under test cannot be established. If only one of the two watermarks are found, the system reports that the video has been altered, and it also reports the track missing the watermark. If the audio is missing the watermark, the system reports the video as fake made by audio impersonation. If both audio and video watermarks are found, the system checks the consistency between them to make sure they contain related VID. Only the 5 least significant bits of the VID decoded from the video need to match the LVID decoded from the audio. If they do not, the system reports to the user that the video is fake. If the consistency check is successful, then the system uses the ASN and VFN numbers decoded from the payload to check whether the audio and video segments are consecutive. Audio segments and video frames need to be consecutive without gaps or repetition; otherwise, the system flags the video as fake. If audio segments and video frames are consecutive, the system proceeds to check whether the ASH and VFH hashes decoded from the payload are the same as those measured in the video itself. If the ASHs of an audio segment are different, the system reports that segment was replaced. If the VFHs of a frame are different, the system reports that the face in that frame has been replaced.

The second stage of authentication uses information embedded in the watermark and information included in the metadata stored in the blockchains. If access to the blockchain is available, the system can perform forensic analysis using the VID decoded from the image watermark found in the video under test. The system uses the VID as an address to access the blockchain, retrieve the corresponding CID, then retrieve the original video from the IPFS and display it to the user. The system can retrieve metadata stored in the blockchain and use it to perform further forensic analysis as following:

1. The system can transcribe the suspected audio and check if the transcription matches the transcription stored in the metadata.
2. The system can detect the shot boundaries in the suspected video and check if they match the shot boundaries included in the metadata,
3. The system can detect the robust key points in the suspect video and compare them with those stored in the blockchain. These key points may be SIFT key points, MFCC coefficients, significant DCT coefficient, locations of peaks in the audio spectrogram, or any other robust features usually used for audio and video fingerprinting.

The system declares the suspected video as fake if it finds a mismatch between any measured feature and the corresponding feature included in the metadata.

8. Conclusions

A system for detecting Deepfakes of news videos was described. The system is based on audio and video watermarking and blockchain technology. The system uses Digimarc robust audio and image watermark technologies. It also uses the IPFS and Ethereum blockchain technologies for storing the video and its metadata, which are used for video forensic analysis at the back end of the social media networks. Proof-of-concept simulations of the main parts of the system were performed, and the preliminary results are encouraging. They indicated that digital watermarking technology can be used successfully to link the video to its original copy and to the metadata stored in a blockchain network. They also indicated that the watermark embedded in the video can be detected after applying Deepfakes. Proper countermeasures for the copy attack were described and should be in place to have an effective system. The system can be generalized to include puppet-master Deepfakes and types of video other than news video.

References

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative Adversarial Nets," in *Proceedings of the International Conference on Neural Information Processing Systems (NIPS)*, 2014.

[2] Deeptrace, "https://regmedia.co.uk/2019/10/08/deepfake_report.pdf," September 2019. [Online].

[3] D. Harris, "Deepfakes: False Pornography is Here and Low Cannot Protect You," *Duke Law & Technology Review,* vol. 17, no. 1, pp. 99-128, 2018.

[4] "Three Threats Posed by Deepfakes That Technology Won't Solve," October 2019. [Online]. Available: https://www.technologyreview.com/s/614446/deepfake-technology-detection-disinformation-harassment-revenge-porn-law/.

[5] F. Matern, C. Riess and M. Stamminger, "Exploiting Visual Artifacts to Expose Deepfakes and Face Manipulations," in *Proceedings of the IEEE Winter Applications of Computer Vision Workshops (WACVW)*, Waikoloa Village, HI, USA, USA, 2019.

[6] M. Koopman, A. M. Rodriguez and Z. Geradts, "Detection of Deepfake Video Manipulation," in *Proceedings of the 20th Irish Machine Vision and Image Processing conference (IMVIP)*, 2018.

[7] X. Yang, Y. Li and S. Lyu, "Exposing Deep Fakes Using Inconsistent Head Poses," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[8] Y. Li and S. Lyu, "Exposing DeepFake Videos By Detecting Face Warping Artifacts," in *Proceedings of Computer Vision and Pattern Recogniton (CVPR)*, 2019.

[9] S. Agarwal and H. Farid, "Protecting World Leaders Against Deep Fakes," in *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, 2019.

[10] Y. Li, M.-C. Chang and S. Lyu, "In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking," in *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS)*, 2018.

[11] D. Guera and E. J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," in *Proceedings of the IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS)*, 2018.

[12] S. Lyu, "Detecting Deepfakes: Look Closely to Protect Against Them," July 2019. [Online]. Available: https://www.salon.com/2019/07/04/detecting-deepfakes-look-closely-to-protect-against-them_partner/.

[13] "Deepfakes Datasets," Kaggle, December 2019. [Online]. Available: https://www.kaggle.com/c/deepfake-detection-challenge/data.

[14] "Deepfake Detection Challenge," [Online]. Available: https://deepfakedetectionchallenge.ai/.

[15] D. M. Turek, "Semantic Forensics (SemaFor)," [Online]. Available: https://www.darpa.mil/program/semantic-forensics.

[16] Truepic, [Online]. Available: https://truepic.com/about-us/.

[17] Serelay, [Online]. Available: https://www.serelay.com/.

[18] "Authenticity Verification of User Generated Video Files," [Online]. Available: https://prover.io/#intro.

[19] R. K. Sharma, B. A. Bradley, S. T. Shivappa, A. Kamath and D. A. Cushman, "Audio Watermark Encoding With Reversing Polarity and Pairwise Embedding". U.S. Patent 9305559, 5 April 2016.

[20] A. R. Gurijala, S. T. Shivappa, R. K. Sharma and B. A. Bradley, "Human auditory system modeling with masking energy adaptation". U.S. Patent 10043527, 7 August 2018.

[21] J. D. Lord, "Watermarking and Signal Recognition For Managing and Sharing Captured Content, Metadata Discovery and Related Arrangements". U.S. Patent 9454789, 29 November 2018.

[22] A. M. Alattar, E. T. Lin and M. U. Celik, "Digital Watermarking of Low Bit-Rate Advanced Simple Profile MPEG-4 Compressed Video," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 12, no. 8, pp. 787-800, August 2003.

[23] A. M. Alattar, E. T. Lin and M. U. Celik, "Digital Watermarking of Low Bit Rate Video". U.S. Patent 8638978, 28 January 2014.

[24] C. Atkinson, "What Are The Types of Metadata Online Video Creators Can Use?," June 2012. [Online]. Available: https://tubularinsights.com/types-metadata-video-creators/.

[25] T. F. Rodriguez and M. M. Weaver, "Robust Encoding of Machine Readable Information in Host Objects and Biometrics, and Associated Decoding and Authentication". U.S. Patent 15/368,635, filed 4 December 2016.

[26] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision,* vol. 60, no. 2, p. 91–110, 2004.

[27] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf and S. Capkun, "On the Security and Performance of Proof of Work Blockchains," in *CCS 2016 - Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Vienna, Austria, October 2016.

[28] H. R. Hasan and K. Salah, "Combating Deepfake Videos Using Blockchain and Smart Contracts," *IEEE Access,* vol. 7, 2019.

[29] M. Kutter, S. Voloshynovskiy and A. Herrigela, "The Watermark Copy Attack," in *Proceedings of SPIE: Security and Watermarking of Multimedia Content II,*, San Jose, CA, USA, January 2000.

[30] J. K. Barr, B. A. Bradley, B. T. Hannigan, A. M. Alattar and R. Durst, "Layered Security in Digital Watermarking". U.S. Patent 8190901, 29 May 2012.

[31] A. Alattar, "Authentication of Physical and Electronic Media Objects Using Digital Watermarks". U.S. Patent 7822225, 26 October 2019.

[32] K. Zhan, Z. Zhang, Z. Li and Y. Qiao, "Joint Face Detection and Alignment Using Multi-task Cascaded Convolutional Networks," *IEEE Signal Processing Letters (SPL),* vol. 23, no. 10, pp. 1499-1503, 2016.

[33] P. Cano, E. Batlle, H. Mayer and H. Neuschmied, "Robust Sound Modeling for Song Detection in Broadcast Audio," in *Proceedings of the 112th AES Convention*, 2002.

[34] "IPFS powers the Distributed Web," [Online]. Available: https://docs.ipfs.io/.

[35] "IPFS versioning - How to get all files from the IPFS key?," [Online]. Available: https://ethereum.stackexchange.com/questions/63109/ipfs-versioning-how-to-get-all-files-from-the-ipfs-key.

[36] "Face Recognition," [Online]. Available: Available: https://github.com/1adrianb/face-alignment.

[37] "DeepFaceLab," GitHub, [Online]. Available: https://github.com/iperov/DeepFaceLab.

Author Biography

Adnan Alattar is a Principal R&D Engineer at Digimarc Corporation. He received his BS in Electrical Engineering from the University of Arkansas (1984) and his MS and PhD in Electrical Engineering from North Carolina State University (1985 and 1989). His fields of interest are Digital Watermarking, Signal and Image Processing. He joined Digimarc Corporation in 1998.

Ravi Sharma is the senior director of research and development at Digimarc corporation. He received his BS in Electrical Engineering from University of Mumbai (1992) and his PhD in Electrical Engineering from Oregon Graduate Institute (1999). Among his areas of interest are Digital Watermarking and Signal/Image Processing. He joined Digimarc Corporation in 1998.

John Scriven is a software Engineer at Digimarc corporation. He received his BS in Computer Science from Oregon State University (2007). His areas of interest are Digital watermarking and Image Processing. He joined Digimarc Corporation in 2017.

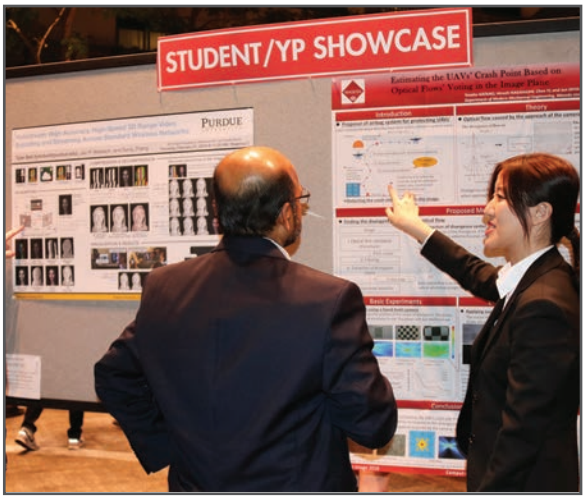
JOIN US AT THE NEXT EI!

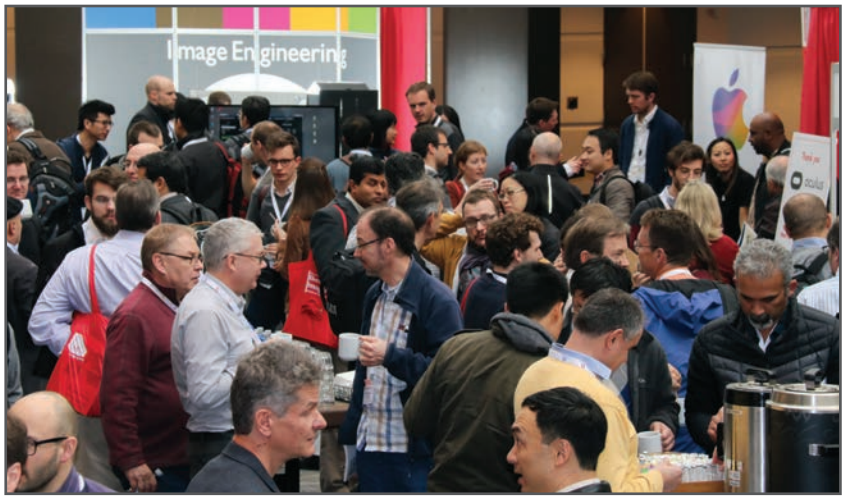
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org