Deepfake Forensics: Identifying Real Regions in Altered Videos with Digital Watermarking

Aayush Asthana[1] and Sam Saarinen[#]

[1]Dougherty Valley High School, San Ramon
[#]Advisor

ABSTRACT

This paper describes a method for detecting Deepfake videos using a lightweight yet secure video encryption algorithm. With the increasing use of digital media, transferring data via the Internet or other mediums requires protection. In the proposed method a digital signature is generated and encrypted using Asymmetric Encryption (RSA). This encrypted signature is then used as a blind watermark for the video. This technique aims to detect “face swap” type of Deepfake videos. It is an efficient algorithm and has minimal impact on the perceptibility of the video quality.

Introduction

Deepfake videos are computer-generated videos that modify or replace the appearance, expressions, scenes, or sounds of actual people or events with those of another person or event using advanced deep learning "models." Deepfake videos get published on websites or social media apps. Advanced deep-learning techniques, like "FaceSwap," are often used to generate these videos. They might use collections of images and videos of real people or events, so it would be impossible to tell the generated "fake" videos apart from the real ones.

Deepfake videos can potentially propagate false information, invade people's privacy, tarnish their reputations, and strain their interpersonal connections. They can also make it difficult to determine whether the information is reliable, which effectively makes the source of the information less reliable.

According to a news article published by Skynet Today in 2020, the number of online Deepfake videos increased by over 600 percent between 2018 (7,900) and 2020 (49,00). In 2019, a Deepfake video of former House Speaker Nancy Pelosi appearing to be under the influence of alcohol was created and circulated on social media. However, before it was declared a Deepfake, it was extensively distributed and received millions of views across various social media sites. The event garnered widespread media coverage, which raised concerns about the potentially detrimental influence of Deepfake videos and, in this instance, the ramifications during political elections. In the same year, Mark Zuckerberg, the CEO of Facebook, appeared in a Deepfake video in which he extolled the virtues of Facebook's possession of the information of billions of users.

Deep learning-powered video-making apps have recently been developed and made freely available online. These tools use Generative Adversarial Networks (GAN), a deep learning approach. A novice user can use them to swiftly change the appearance, pronunciation, or speech and create videos that look genuinely authentic. The sophistication with which Deepfake movies are created enhances the challenge of identifying and detecting such fake videos.

Researchers have proposed various Deepfake detectors. These include detecting "blinking rate" anomalies, inconsistencies in head angle, and faults in changing movies such as "ghost edges," "edge blurring," or "skin tone variation." Though these AI-trained models can detect and identify Deepfake with some accuracy, as advanced deep learning models are invented, these detectors become less accurate and require additional training data. Another method that is being investigated is the use of video encoding techniques during video production. These methods employ a combination of encryption and digital watermarks that are embedded in video frames. If a Deepfake video

is created from these watermarked videos, it does not include the identical " hidden " signatures in the original video. This is an excellent technique to warn the user if the original video was tempered.

This work discusses a novel signature registering approach that uses RSA Private-Public Key Security to generate a digital signature and embeds it in the video such that it retains video perceptibility and aids in identifying and preventing Deepfake. It can quickly confirm any breach of the video's integrity.

What is a Digital Watermark?

Watermarking techniques, also referred to as digital signature, sign images by introducing changes that are imperceptible to the human eye but easily recoverable by a computer program.

Different Types of Digital Watermarks

In a blind watermarking scheme, neither the original video nor the embedded watermarks are required for detection but just the secret keys. In a semi-blind watermarking scheme, only some information from the original cover and the secret keys are needed. While in a non-blind watermarking scheme requires the original cover, the original watermark, and the secret keys are required.

Popular approaches to Digital Watermarks

- Extended image watermark technique to all frames of the video
- Video is initially divided into video shots. Then from each video shot, one video frame called an identical frame is selected for watermark embedding.
- Select the frames with the biggest luminance value in every shot to be the host frames.
- Frames are selected based on scene change detection.

Different Types of “Attacks” on Digital Watermarks

An” Attack” is an intentional or unintentional alternation of signed videos such that it would be possible to retrieve the signature. There are attacks that focus on the entire video, like -frame dropping, frame inserting, and frame rate changes, which are termed Temporal attacks. Attackers might try to obtain multiple watermarked data without knowing the watermarking algorithm and remove the watermarks.

Challenges in Digital Watermark techniques

Imperceptibility

The algorithm embeds the watermark in video frames so that the quality of the video frames is not perceptually affected. Due to the temporal (sequence of frames) nature of a video, embedding digital watermarks could create distortions within or across frames.

Robustness

It is the ability of the algorithm to extract the watermark successfully from the video. The video (or the frames) could be degraded by various attacks. Often the algorithm needs to be immune to added noise (maybe unintentionally added in transmission) or media compression algorithms like JPEG or MPEG.

Performance

The complexity of the watermarking scheme should be low because of the significant number of frames to be processed in a video signal.

What is a Video Cryptography?

Video Cryptography involves securing video content through various encryption and cryptographic techniques. The main goal of video cryptography is to protect the confidentiality, integrity, and authenticity of video data, ensuring that only authorized parties can access and view the video content while preventing unauthorized access. Video cryptography typically involves applying cryptographic algorithms ("hash functions") to an input video file to either encrypt or decrypt a video.

Different cryptography signature generation algorithms

- Message Digest 5 (MD5): It is a widely used hash function that produces a fixed-size hash value (128-bit) from input data.
- Secure Hash Algorithm (SHA): It is a family of cryptographic hash functions (e.g., SHA-1, SHA-256, SHA-3), used for generating fixed-size hash values.
- Digital Signature Algorithm (DSA): It is a public-key digital signature algorithm that enables the creation of digital signatures to authenticate the origin and integrity of digital data.
- Elliptic Curve Digital Signature Algorithm (ECDSA): It is a variant of DSA that provides similar digital signature capabilities with smaller key sizes, making it more efficient in terms of computation and bandwidth.
- Rivest-Shamir-Adleman (RSA): It is a widely used "asymmetric" encryption algorithm. It's based on creating a pair of key - public key and private key. Public keys is openly shared, while the private key is kept confidential with the owner.

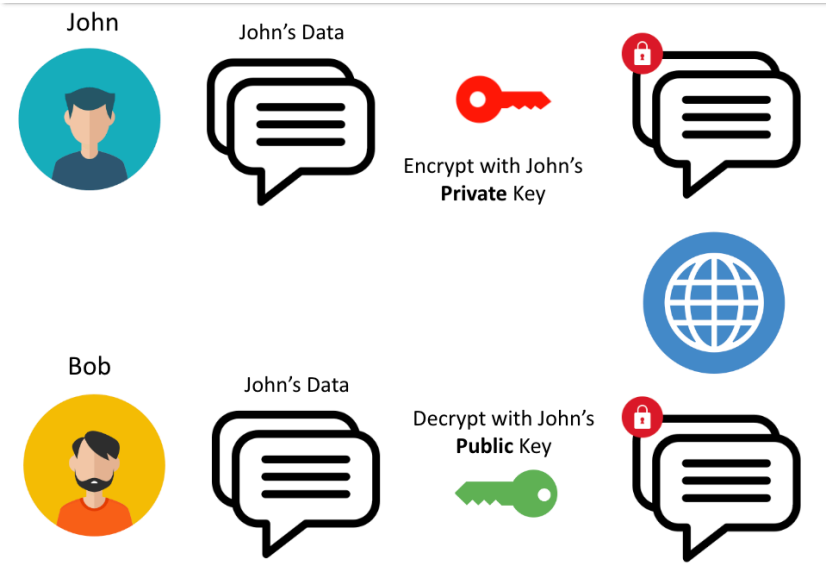

Figure 1. Asymmetric Encryption: As seen in Figure 1, data could be encrypted using a private key and encrypted data could be shared over the internet. This data could be decrypted using a well-known (published) public key.(Photo credits: https://www.flaticon.com/authors/freepik)

General approach to video cryptography

Creation and embedding of signature.

- Generate a signature by providing data from the video frames as an input to the cryptography algorithm.
- Embed the signature in the video itself such that the video quality is not impacted. (like 'digital watermarking')

During the verification phase

- Signatures are extracted from the video frames. Extracting the signature will require decryption (corresponding to the encryption algorithm)
- Signatures are also generated on the corresponding frames.
- Extracted signatures are compared with the generated signatures. If they are found to be the same, then it is safe to assume that the video is not modified.

Method

Detecting the "Area-of-Interest"

A typical video is made of multiple frames. Each frame has a fixed dimension (height and width) expressed in numbers of pixels. For example, DFDC dataset has 256 x 256 frame size. Pixels are 3-byte values, where each byte represents the 3 color channels – Red, Green & Blue (RGB) respectively. So, each color channel could have a value between 0 and 255. To generate a signature, each frame is further divided into equal sized square "tiles" with a dimension 'd' pixel. For example, for d = 48, a tile will be 48 x 48 pixels in size.

Most of the Deepfake videos are generated by modifying facial area (like – 'face swap') in the video. Hence, the set of all the tiles that make up the human face in the video is considered the "Area-of-Interest" of the entire video. These tiles are the most important to detect Deepfakes. Other 'background' tiles (tiles other than face area) could also change, however some of these changes could be permissible (e.g. adding a color filter). Furthermore, this helps in faster performance of the algorithm, and lesser impact on the perceptibility of the video (after the signature has been embedded).

MTCNN (Multi-Task Cascaded Convolutional Neural Networks) is a neural network model that helps in detecting the face boundaries and important landmarks of a human facial area. A python implementation (*facenet in pytorch*) of MTCNN is used to detect the face in each video frame. After the area of the face is detected, tiles that constitute the face are calculated from each frame of the video.

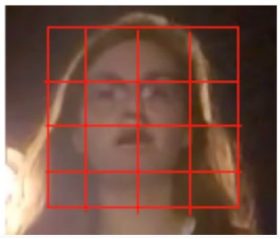

Figure 2. "Area-of-Interest". As seen in Figure 2, MTCNN neural network is detecting the face of a human in each frame, the box area of face detection is then matched with the tiles of the frame to constitute the "Area-of-Interest". (Photo credit: https://www.kaggle.com/competitions/Deepfake-detection-challenge/data)

This set of tiles ("Area-of-Interest") is fed into the next phase of signature generation and embedding algorithm.

Signature generation and embedding

This processing is performed at the source or at the place of video generation. The "Area-of-Interest" list of tiles are used for the generation of watermark involves the following steps –

- Creation of the signature: For each tile the mean value (arithmetic average) of each color channel value of the "inner pixels" (1-pixel wide border pixel is left for embedding. Rest of the pixels constitute the inner pixels) is calculated. Next, all three-color mean values, the frame number and tile number are combined to form a signature.
- Encrypting the signature: Unique public and private keys are generated using the RSA algorithm. These keys are used to encrypt the signature and create 32 bytes encrypted signature.
- Embedding the signature into the video: For each tile the middle 32 pixels of the top row is where the signature is embedded. Each signature byte (8 bits) is divided into 3-bits, 3-bits and 2 bits. The top 3 bits are stored in the least significant 3-bits of the red color value. The mid 3 bits are stored in the least significant 3-bits of the green color value and bottom 2-bits are stored in the least significant 2 bits of blue color value.
- In a worst-case scenario, only red and green color values would change by 8 (for 3-bits) and the blue color value would change by 4 . This scheme has minimal impact on perceptibility of each tile when embedding each tile with the encrypted signature.

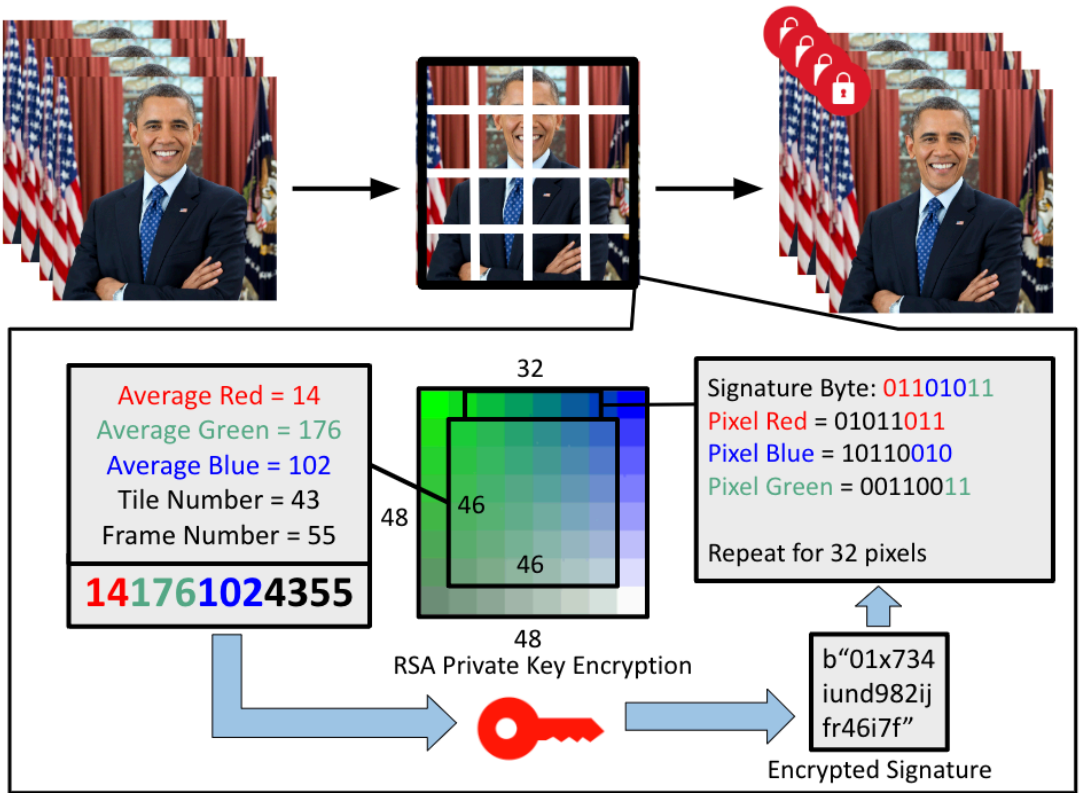

Figure 3. Signature Creation and Embedding Scheme. As seen in Figure 3, signature is created using the mean value of all color channels of the inner pixels and is combined with the tile number and frame number. RSA encryption is used to generate a 32-bit signature watermarked in the top row of tile. (Photo credit: www.buzzfeed.com/craigsilverman/obama-jordan-peele-Deepfake-video-debunk-buzzfeed)

Signature verification process

This process is performed at the target or at the place where the video is published. Verification of signature involves the following steps:

- Decryption of the signature: Each video frame is divided into equal sized tiles. RSA decrypt is applied to the top-central 32bytes of each tile. This will result in extracting the original signature that was watermarked in each tile.
- Generations of the signature: Each video frame is divided into equal sized tiles. The mean value of each color channel value is calculated. Next, all three-color averages, the frame number and tile number are combined to form a unique signature.
- Comparison the signature: The extracted and generated signature are compared to verify a match or a mismatch.

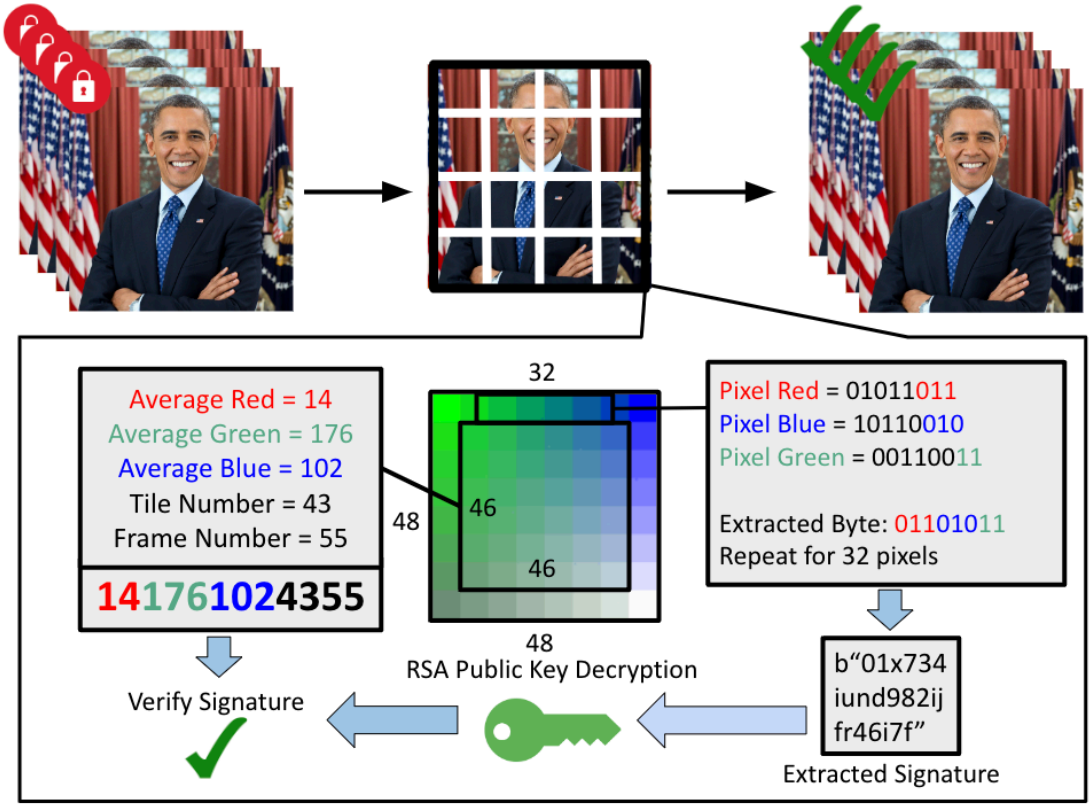

Figure 4. Signature Extraction and Verification. As seen in Figure 4, signature is extracted from each tile of each frame, decrypted using the public key and verified against the calculated signature of the same tile. (Photo credit: www.buzzfeed.com/craigsilverman/obama-jordan-peele-Deepfake-video-debunk-buzzfeed)

Results

Data used for measurement.

To counter the emerging threat of Deepfake, Meta (Facebook) has constructed a huge face swap video dataset to enable the training of detection models and organized the accompanying Deepfake Detection Challenge (DFDC) Kaggle competition. The DFDC dataset is by far the most extensive publicly (and freely) available face swap video dataset. Each video clip is 1080p resolution and resized to 256 x 256 frame size. For each video clip, metadata would specify if the video were fake, and if the video is fake, what is the corresponding real video. The dataset can be freely downloaded from the Kaggle competition website.

After downloading the data, following additional steps were taken.

1. Videos that DFDC metadata marked as "real" belong to the R set.
2. Videos that DFDC metadata marked as "fake" belong to the F set.
3. R set of videos were then signed and encrypted (using the above-mentioned technique) and were marked as R' ("R prime")
4. F set of videos were then signed and encrypted (using the above-mentioned technique) and were marked as F' ("F prime")

This helped to simulate real world scenarios. R' represents a signed video from the publisher, then the corresponding F video would fail to decrypt. This would be flagged as a warning with invalid signature. While if R' is compared with F', the signature would be extracted but will not match with the R' representing a fake video.

Perceptibility

To what extent does the video's quality suffer because of the watermark? Ideally, the watermark wouldn't have a major impact on the video's sound or picture quality when it was added. To preserve the viewing experience and prevent the watermark from detracting from the content itself, it is essential that watermarks be nearly undetectable. To measure the impact of watermarking on the perceptibility of the video, the maximum change is the color value of each pixel is calculated. Since there are 3 bits used, the maximum color value change would be 7 (0-8). Hence the Perceptibility could be expressed as:

$$\mathcal{P} = \frac{\sigma}{d^2} \cdot \frac{(2^{\delta} - 1)}{255 \cdot 3}$$

where ,
σ = number of bytes in the signature
δ = number of bits used to encode the signature
d = dimension of the tiles
$\mathcal{P}$ = Perceptibility percentage

For a tile size of 48 x 48, the perceptibility percentage would ~ 0.03%. In other words, only 0.03% of color value would change per tile. When we start considering only the tiles in the "Area-of-Interest", this change would be further insignificant.

Algorithm Time Complexity

As the number of frames increases the time complexity of the algorithm increases. Also, if the time complexity of the algorithm is inversely proportional to the dimensions of the tiles. If the size of the tile is larger then lesser number of tiles will need to be processed.

If,

$n = number\ of\ frames\ in\ the\ video,$
$w = width\ of\ the\ frame,$
$h = height\ of\ the\ frame,$
$d = dimension\ of\ the\ tile,$
$\omega(n) = time\ to\ watermark = \ 2.w.h\ (n/2d + n)$
$\Delta(n) = time\ to\ verify = \ 2.w.h\ (n/2d + n)$
$Big\ O\ complexity\ = \ O(\frac{n}{d})$

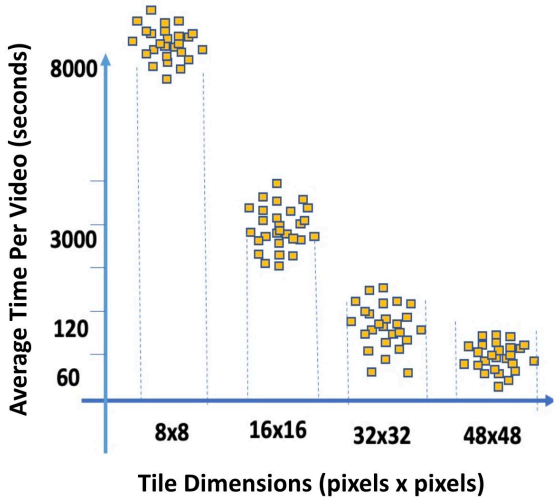

Figure 5. Tile dimension vs Time complexity. As seen in Figure 5, as the tile decreases the complexity increases in a curve like f(x) = 1/x as the Big-O complexity is O(n/d)

Detection of mismatched tiles

Experiments looked at all the tiles of each frame of each fake video and compared them with the real videos. Even though there was no perceptible change in the fake videos to ordinary eyes, there were several changes in the fake videos at a pixel level. On average, ~60% to 80% of the tiles of the entire video had changed for a fake video. These changes could be anywhere in the frame – sometimes a tiny background lighting change.

However, when "Area-of-Interest" tiles were considered, this number significantly increased from ~80% to ~100%. It is a classic hallmark of deepfake videos – adding subtle changes in each frame that human eyes cannot detect as fake. The DFDC dataset has been created to test the algorithms with extreme cases -where the changes are very small.

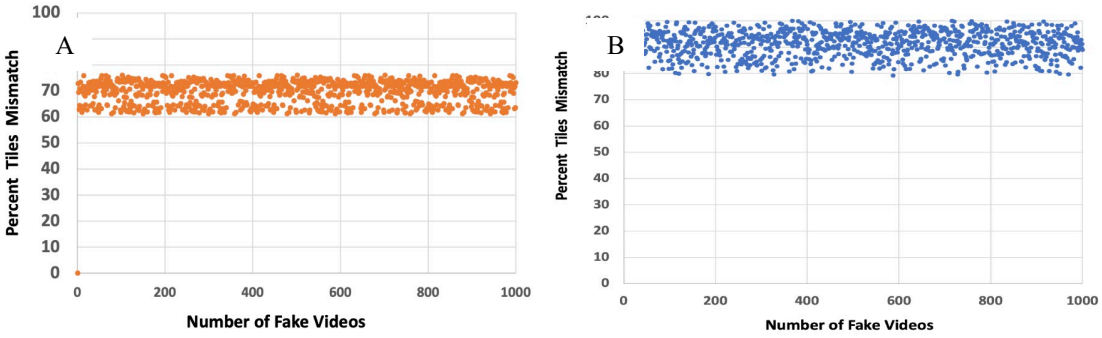

Figure 6. Percentage of mismatched tiles in fake videos : As seen in Figure 6A, all the tiles of all frames are considered for checking for mismatch. While in Figure 6B, only the "Area-of-Interest" tiles are considered for checking for mismatches.

Source code

The algorithm is developed using Python and its various package. The entire source code can be found at https://github.com/aayushasthana/Deepfake-Detection

Discussion

In the early days of the internet's expansion, a pressing issue emerged as users ventured into unfamiliar online territory. The concern revolved around the security risks of visiting unfamiliar websites. This solution came in the form of signed certificates, a digital mechanism that brought a sense of trust to the online world.

To tackle this problem, specific companies were responsible for verifying website authenticity. Upon successful authentication, these companies would issue digital certificates. These certificates acted as a safeguard against potential cyber threats, employing the principles of public-private key encryption to establish a secure connection between users and the websites they visited.

Fast-forward to today, deepfake videos, intricately engineered to simulate real individuals and scenarios convincingly, have given rise to security concerns. To address these concerns of deepfake, an innovative concept emerges, drawing inspiration from the solution that quelled the uncertainties of the internet's early days.

Imagine a scenario where the creators of videos—individuals, organizations, or content creators—could register their creations with a dedicated provider. This provider's role would mirror that of the certificate authorities, authenticating the videos and issuing certificates that vouch for their integrity. Just as a company's legitimacy lent credibility to a website, these certificates would serve as digital markers of trustworthiness for videos.

To implement this vision, an RSA encryption key could be used. Once authenticated by the provider, the videos would undergo the transformative process of being signed with this encryption key. Much like the digital certificates of the past, this signature would serve as a virtual seal, attesting to the video's authenticity and origin.

Imagine a world where online video platforms, equipped to interact with these certificates, could not only display the videos but also provide viewers with insights into their reliability. Just as a cautious internet user would gravitate towards websites with established certificates, viewers could gauge a deepfake video's credibility based on the strength of its certificate.

Essentially, the solution here refers to the foundational principles that fortified the internet's growth—authenticity, trust, and encryption. By applying the lessons learned from securing the digital landscape to the problem of deepfake videos

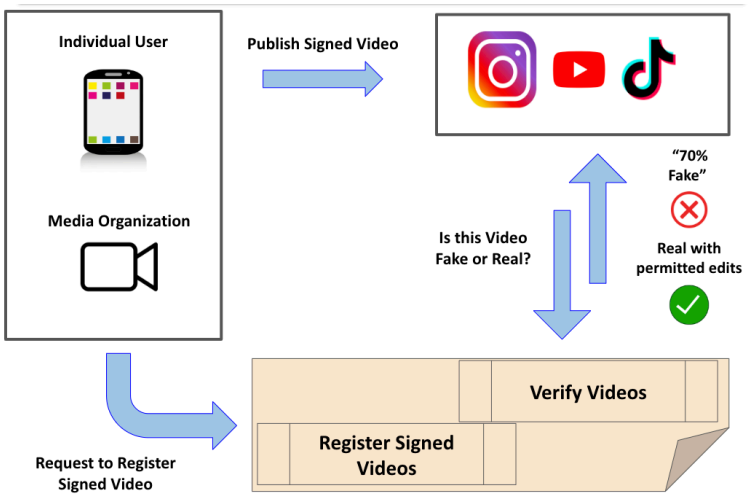

Figure 7. A proposed real world implementation: As seen in Figure 7, the publisher of the video would register with certificate provider and will be provided with the encryption key. Either a software running on a smart phone or a specialized chip could perform the encryption and signature embedding and publish the video. The viewing website could connect to the certificate provider when the video gets viewed for its authenticity.

Conclusion

The algorithm demonstrates high accuracy in identifying and pinpointing deepfake videos; particularly those forged using FaceSwap, showcasing high accuracy. The foundation of the algorithm is built upon a robust RSA-based encryption framework, providing excellent protection from any attempts to tamper with or extract the watermark. The proposed watermark embedding method has a mere 0.03 visual impact on the video. This imperceptible alteration ensures that the original content remains intact while the watermark discreetly fulfills its authentication role. The algorithm is highly performant by efficiently encrypting and decrypting frames at an impressive pace of approximately 0.15 seconds per frame.

Limitations

The proposed solution offers potential for improvement. There have been documented deepfake videos in which the audio component has been modified, but the visual data in each frame has remained unchanged. There is potential to extend digital watermarking to include audio tracks extracted from the source video, making the system more robust. The algorithm prohibits the users from changing the video, such as brightness changes or backdrop substitutions. The algorithm might take a more flexible approach, allowing the video's publisher to make permissible alterations while maintaining the critical components of the video. At this time, because the signatures are generated separately for each tile within each frame, the method cannot detect frame drops. A proposed algorithm improvement would be chain the signature across frames thereby help in detecting frame drops.

Acknowledgements

I thank my advisor for his invaluable insights and guidance throughout this topic. His expertise and thoughtful perspectives have been instrumental in shaping my understanding. I appreciate all the engaging discussions that fueled

my exploration of ingenious problem-solving approaches. His unwavering support and mentorship have made this endeavor educational and genuinely enriching.

References

"A deepfake video of Mark Zuckerberg presents a new challenge for Facebook" https://www.cnn.com/2019/06/11/tech/zuckerberg-deepfake/index.html

"Doctored Pelosi video highlights the threat of deepfake tech" https://www.cbsnews.com/video/doctored-pelosi-video-highlights-the-threat-of-deepfake-tech/

"The State of Deepfakes in 2020"

DFDC Dataset: https://www.kaggle.com/competitions/deepfake-detection- challenge/data

Dolhansky, Brian, et al. "The deepfake detection challenge (dfdc) dataset." arXiv preprint arXiv:2006.07397 (2020).

Haliassos, Alexandros, et al. "Lips don't lie: A generalisable and robust approach to face forgery detection." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021. https://www.skynettoday.com/overviews/state-of-deepfakes-2020

Masood, Momina, et al. "Deepfakes Generation and Detection: State-of-the-art, open challenges, countermeasures, and way forward." Applied Intelligence (2022): 1-53.

MTCNN PyTorch: https://www.kaggle.com/code/timesler/guide-to-mtcnn- in-facenet-pytorch

Prabhishek Singh, R S Chadha, "A Survey of Digital Watermarking Techniques, Applications and Attacks", International Journal of Engineering and Innovative Technology (IJEIT), Volume 2, Issue 9, March 2013