“Is this my president speaking?” Tamper-proofing Speech in Live Recordings

Irtaza Shahid, Nirupam Roy
University of Maryland, College Park
{irtaza,niruroy}@umd.edu

ABSTRACT

Malicious editing of audiovisual content has emerged as a popular tool for targeted defamation, spreading disinformation, and triggering political unrest. Public speeches and statements of political leaders, public figures, or celebrities are particularly at target due to their effectiveness in influencing the masses. Ubiquitous audiovisual recording of live speeches with smart devices and unrestricted content sharing and redistributing on social media make it difficult to address this threat using existing authentication techniques. Given public recordings of live events lack source control over the media, standard solutions falter. This paper presents *TalkLock*, a speech integrity verification system that can enable live speakers to protect their speeches from malicious alterations even when the speech is recorded by any member of the audience. The core idea is to generate meta-information from the speech signal in real-time and disseminate it through a secure QR code-based screen-camera communication. The QR code when recorded along with the speech embeds the meta-information in the content and it can be used later for independent verification in stand-alone applications or online platforms. A user study with live speech and real-world experiments with different types of voices, languages, environments, and distances show that *TalkLock* can verify fake content with 94.4% accuracy.

CCS CONCEPTS

• **Security and privacy → Authentication**; **Tamper-proof and tamper-resistant designs**; **Usability in security and privacy**.

KEYWORDS

Deepfake; Speech verification; Voice features; QR code

ACM Reference Format:
Irtaza Shahid, Nirupam Roy. 2023. “*Is this my president speaking?*” Tamper-proofing Speech in Live Recordings. In *The 21st Annual International Conference on Mobile Systems, Applications and Services (MobiSys ’23), June 18–22, 2023, Helsinki, Finland.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3581791.3596862

1 INTRODUCTION

DeepFakes – a common term for intentionally deceptive synthetic audiovisual content – have emerged as a primary tool for spreading disinformation. Recent research works [70] and clips [25, 51] largely circulated on social media underscore the capabilities of deepfakes where influential people, including former US presidents Barack Obama and Donald Trump, appears to deliver misleading or profane statements. While the specialized equipment and substantial library of training content required for deepfakes prevent its widespread usage by amateurs [34, 62], “shallow” fakes represent a more immediate threat. ShallowFakes are commonly available audiovisual clips that are selectively edited or manually altered and recirculated to spread confusion and mislead an audience. For instance, doctored footage of the U.S. House of Representatives speaker Nancy Pelosi surfaced where she appears to be intoxicated and slur her words during a press conference [49]. Audiovisual media being the most persuasive form of content on social platforms and news [36], synthetic videos can manipulate the masses for political gains as seen in the case of the fake video depicting Ukrainian president Volodymyr Zelenskyy appearing to tell his citizens to surrender the fight against Russia. It is considered one of the greatest threats to democracy [58] for widespread implications in political slander and defamation, propaganda, and manipulating social attitudes. Unfortunately, despite tremendous effort, techniques to protect and verify the authenticity of audiovisual content is sparse and non-existent for public recordings of live events. We ask the question – *Is it possible to protect a publicly delivered speech from alteration even when anyone from the audience can record it live and publish it?*

Existing methods for detecting altered audiovisual media fundamentally adopt one of the two strategies: (a) artifact detection and (b) embedding and verification of meta-information. The first method seeks to identify any subtle abnormality in the sound or video that might have been introduced during the editing process. A class of techniques focus on the explicit technical inconsistencies generated during editing operations like warping [42], blending [39], and splicing [33] or signatures left by the commonly used learning models [65, 68, 71, 72]. Latest detection algorithms extend artifact detection to the human factors, such as minute mismatch in spoken words and lip movements [12, 28], unnatural eye blinks [41], inconsistent head poses [67], facial blood flow, heart rate [31] or emotional state [47] of the speaker. However, the artifact-based approaches rely on the limitations in the present-day alteration techniques and therefore offer only a short-term solution until these techniques evolve to produce more sophisticated fake content. This results in an arms race, which is decidedly in favor of undetectable fake content production [11, 30].

The second class of solutions, on the other hand, relies on adding specific signatures or meta-information to the media file for content integrity verification. It leverages control over the information

MobiSys ’23, June 18–22, 2023, Helsinki, Finland

ACM ISBN 979-8-4007-0110-8/23/06.
https://doi.org/10.1145/3581791.3596862

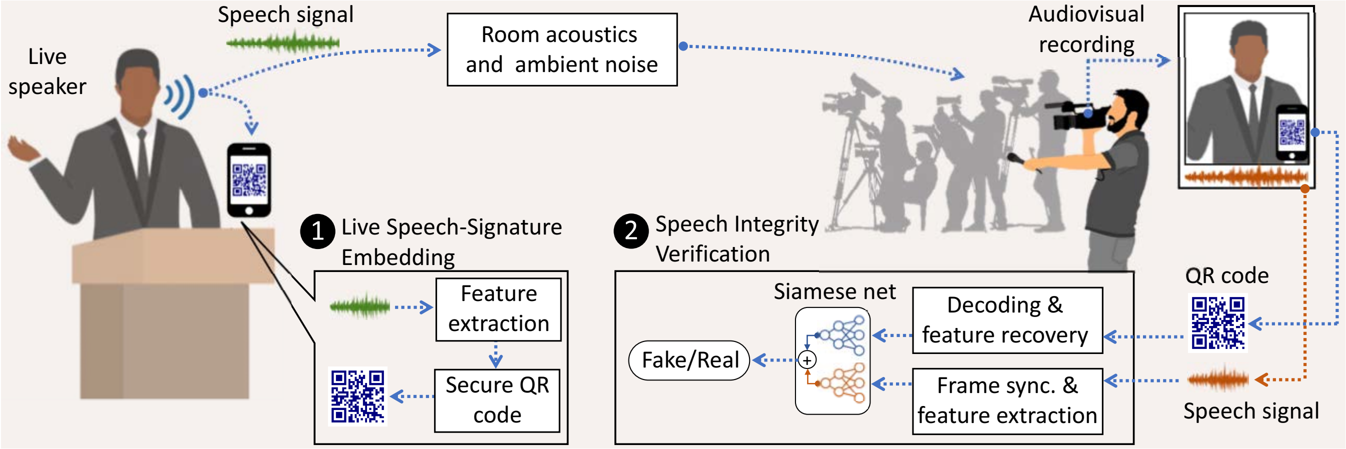
Speech signal
Live speaker
Room acoustics and ambient noise
Audiovisual recording
1 Live Speech-Signature Embedding
Feature extraction
Secure QR code
2 Speech Integrity Verification
Siamese net
Decoding & feature recovery
Frame sync. & feature extraction
Fake/Real
QR code
Speech signal

Figure 1: Design overview: ❶ The live speech-signature embedding module extracts features from speech signals in real-time and generates a sequence of cryptographically secure QR codes. ❷ The speech integrity verification module uses our algorithm to check the speech in the content under question matches with the features recovered from the QR codes visible in the video.

at the source and embeds immutable signatures or markers on the media. Available techniques range from adding obfuscated information or digital watermarks to the content to introducing identifying data along with the media information [43, 48, 55]. The Content Authenticity Initiative [3] strives to develop a standard to enforce meta-information-based provenance that can also be effective against fake contents [64]. This approach can leverage strong cryptographic techniques to assure efficient authenticity verification. Naturally, these approaches require access to the content at source or cooperation from the recording system, which limits its application to highly controlled information publishing methods. For instance, a pre-recorded public statement can include a verification code to the file's header before releasing it to the public or news channels. However, such techniques are largely ineffective for a live public speech which can be recorded and distributed by anyone in the audience. Smartphones have given the capability of recording to almost every individual and social platform has provided access to copy, alter, and redistribute content. While this underscores the need for protecting live speeches more now than ever, existing techniques cannot bridge the gap.

In this paper, we present *TalkLock*, a speech integrity verification system – the first of its kind that can enable live speakers to protect their speeches from malicious alterations even when the speech is recorded by any member of the audience. The core idea is to generate real-time meta-information from the live speech signal and disseminate it in a way that it will be included in the recording. In the verification stage, the system expects matching features from the audio data in the recording with this meta-information.

An end-to-end use case scenario is shown in Figure 1, where the speaker's mobile phone or a tablet serves as the ❶ **live speech-signature embedding module**. This module 'listens' to the live speech and continuously generates cryptographic codes with carefully designed verifiable features of the speech. The device's screen faces the audience and the sequence of codes is shown on the screen as a sequence of QR codes. The QR codes are captured in the audio-visual recording along with the speech – basically embedding two information streams; the audio data of the speech and the cryptographic code sequence in the video frames. The presence of the QR code on the recording marks the recording as a *TalkLock* verifiable content. The speech in the audio-visual recording remains verifiable even when posted in different formats, on any social media platform, or even when shown on television. Any user, interested in verifying the speech integrity of the content, invokes the ❷ **speech integrity verification module** with the content. This module systematically extracts the features from the audio data and checks against the code sequence to identify if any part of the speech is altered from the original live version.

While the concept of the system is clear, it needs to address several theoretical and practical challenges to be useful in real-life scenarios. First, the speech signal captured by the embedding module (placed near the speaker) is not the same as the signal recorded by the audience due to multipath distortions and environmental noise. The system needs to identify audio features that are immutable to this channel distortion but at the same time sensitive to any subtle alteration made by the attacker. Next, the system should prevent any possible manipulation of the captured QR code sequence, including opportunistic reuse or reordering of the code that can theoretically dodge verification. Moreover, the system essentially creates a form of screen-camera communication that requires careful design to maximize the usage of the limited data rate and adverse channel effects. *TalkLock* overcomes these challenges to develop the methods and a working prototype of the system.

It is possible to explore variations and improvements in various aspects of the system design in response to specific environments and application scenarios, including the code dissemination modalities, encryption algorithms, and code verification approaches. However, if the core idea is successful, this system can introduce a new genre in the content verification system, i.e., live content verification. It can be effective in protecting leaders and public figures from humiliation or targeted extortion. *TalkLock* can also provide a trustworthy forensic technique to verify recorded live speeches without any cooperation or assumption from the recording device. Moreover, online news and social media can efficiently identify and eliminate manipulated content that can potentially lead to political and social unrest. It can be a step against the growing culture of targeted disinformation.

Summary of contributions:
In developing *TalkLock*, we have made the following contributions:

- *TalkLock* presents a novel tamper-proofing system for live speeches that allows anyone in the audience to record verifiable speech videos.
- A design of two types of environment robust features: temporal energy modulation and time-frequency convolutional features which are robust channel impairments and can identify even minute differences across speech phonemes.
- A channel-agnostic feature matching technique using Siamese network and channel normalization method. This allows us to match features computed from similar audio passed through different channels.
- We implement the prototype and evaluate it with a live speech from users and in different types of scenarios such as different types of voices, languages, environments, and distances.

2 OVERVIEW

We present *TalkLock*’s threat model assumptions and use case scenarios before enumerating the system design.

2.1 Adversary Model

We focus on adversaries whose goal is to maliciously modify or alter speech recordings and redistribute the audiovisual contents. The speaker in the original content can have it recorded in a private setting or in a studio to be released later or can deliver the speech directly to the live audience. In the case of a live speech, we expect any person from the audience, including news reporters, adversaries, and common people can record it with the purpose of legitimate or malicious sharing. While the distribution of malicious content generally surfaces on social media or other online groups, we also consider possible forms of offline digital sharing (e.g., individual file sharing for extortion or use as evidence in court). The adversary may obtain the original content by directly recording the live event or downloading the published legitimate version of it.

TalkLock aims to verify the integrity of the speech in audiovisual content. It is easier for the adversary to imperceptibly alter speech in a video given widely available tools and therefore such threats are immediate and practical with significant impact on the masses. While it is technically possible to extend the proposed model to detect malicious video tampering as well, in this paper we limit our scope to the speech in the video. We expect that the adversary will take necessary actions to modify portions of the video, such as the lip motions so that the change in the speech is impossible to detect with existing techniques.

Our system assumes the dynamic QR code generated by the speech signature embedding module is visible on the video and decodable. However, it is natural for an adversary to attempt to tamper with or modify the QR code in some way. As described in Sections 3 and 4, our system, *TalkLock*, relies on a cryptographic signature to detect compromised QR codes. The cryptographic signatures signed by the private key of the speaker ensure that no adversary can generate, tamper or modify an authentic QR code. This prevents an adversary from launching a sophisticated replay attack by recording a maliciously generated fake speech in the speaker’s environment and generating the corresponding QR codes. In our target scenario, the user delivers a public speech without any expectation for privacy, and *TalkLock* does not introduce any directly identifiable information. However, it is relevant to mention in this context that a resourceful attacker can potentially make use of any features extracted from the user’s voice data, in this case, *TalkLock*’s feature set, to infer sensitive information, such as HRTF, bone conductivity, vocal track features, etc. It is a growing concern with any form of human-centric data and calls for an involved treatment. We aim to focus on this concern in our future work.

It is important to note that, other than the private key of the speaker used in signature embedding and weights of the Siamese network used for verification, we do not assume any information about the system design is kept secret from the adversary. We expect the adversary to be knowledgeable about our feature set, signature packet format, signature embedding algorithm, and every detail of the verification process.

2.2 Design Overview

TalkLock design has two primary modules.

❶ The **live speech-signature embedding module** extracts features from speech signals in real-time and generates a sequence of cryptographically secure Quick Response (QR) codes. This module runs independently on a computing device that has a microphone for recording sound and a screen for displaying the QR code sequence. Any common smart device (e.g., tablet or a smartphone) or a laptop can serve this purpose, or a dedicated sound processing and display module can be developed. A user who wants to protect against malicious alteration of her speech places this module with the QR code screen visible to the audience so that the audiovisual recordings capture the dynamically updated codes. *TalkLock* carefully generates the speech features that remain immutable to the ambient noise and multipath channel propagation, as explained in Section 2.3. These features, along with cryptographic signatures, get embedded in the live recordings of the event through the QR codes.

❷ The **speech integrity verification module** uses our algorithm to check the speech in the content under question matches with the features recovered from the QR codes visible in the video. This module can run on any computing platform, including standalone computers or smart devices and online social platforms. For fully automated verification scenarios, the verification process would require an internet connection to obtain the public key of the speaker. Note that the speech quality in the content can naturally differ from the source sound signal seen by the signature embedding module for the reasons elaborated in Section 3. *TalkLock* verification algorithm is designed for robust integrity check despite these unavoidable differences in a live recording. We elaborate on the process in Section 4.

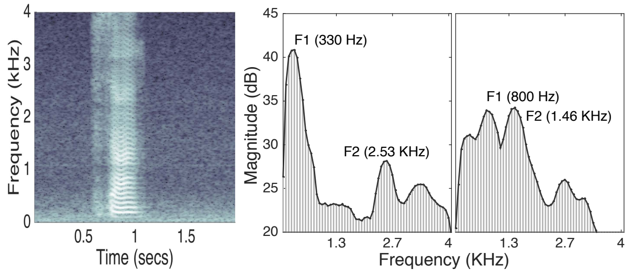

Figure 2: a) The spectrogram of the spoken consonant 's' followed by the vowel 'a' recorded with a microphone, b) The locations of the first two formants (F1 and F2) for the vowel sound *'i'* and *'a'*.

2.3 Primers and Challenges

This section is a high-level introduction to the characteristics of human speech, followed by acoustic phenomena that can impact our feature selection and verification process.

A. Human Speech Basics

Speech production

Human speech can be viewed as periodic airwaves produced by the lungs, modulated through a sequence of steps in the throat, nose, and mouth. When the vocal cords are constricted, the vibrations induced in the airflow are called *voiced* signals. The voiced signals generate high energy pulses – in the frequency domain, the signal contains a fundamental frequency and its harmonics. On the other hand, when the vocal cords dilate and allow the air to flow through without heavy vibrations, the outcome is called *unvoiced* signals. Both voiced and unvoiced signals then pass *glottis*, which further pulsates to add temporal variation to the signal power as well as distinctiveness to different words and an individual's voice.

Structure in speech signals

While the above discussions present a biological/linguistics point of view, we now discuss how they relate to the recorded speech signals and their structures. Figure 2.a shows the spectrogram when a human user pronounces the alphabets "sa" – the signal was recorded through a smartphone microphone [1]. Although a toy case, the spectrogram captures the key building blocks of speech structure. We make a few observations that will underpin the challenges in identifying distinct features in human speech.

(1) The first visible signal (between 0.6 and 0.75 seconds) corresponds to the *unvoiced* component, the consonant "s". This signal is similar to noise with energy spread out rather uniformly across the frequency band. The energy content in this signal is low to moderate.

(2) The second visible signal corresponds to the vowel "a" and is an example of the *voiced* component. The signal shows a low fundamental frequency and many harmonics all the way to 4KHz. Fundamental frequencies are around 85–180Hz for males and 165–255Hz for females [60]. The energy content of this signal is far stronger than the *unvoiced* counterpart.

(3) Within the *voiced* signal, the energy content is higher in the lower frequencies. These strong low-frequency components determine the intelligibility of the spoken phonemes (i.e., the perceptually distinct units of sound [63]) and are referred to as *formants* [37]. The first two formants (say, $F1$, $F2$) remain between 300–2500Hz and completely form the sound of the vowels, while some consonants have another significant formant, $F3$, at a higher frequency. Figure 2.b shows examples of 2 vowel formants – "i" and "a" – recorded by the microphone.

B. Spatial Diversity of Sound Quality

In essence, *TalkLock* attempts to compare speech recorded at two different places – first, the original speech recorded with the embedding module placed near the speaker, and, second the audience records the live speech using different audiovisual recording devices. Speech signals in these two recordings can be significantly different for two major reasons:

(a) Multipath distortion. The major impact is introduced by the multipath propagation of sound waves. Multipath is a natural phenomenon where a signal, after leaving the source (i.e., the speaker's mouth), reflects off objects in the environment to create echoes or replicas. The replicas then propagate through paths of different delays before combining at the receiver (i.e., the recording microphone). The lengths of these individual paths decide the phase delays of the replicas and therefore their superimposition leads to a specific amplitude and phase of the received signal. While this phenomenon is useful for spatial sensing [13, 14, 26], it presents a critical challenge in matching speech signals for verification. Recording device placed at different locations, even when separated by only a few inches, receives a unique combination of path lengths resulting in different signal qualities [57]. Given the embedding device is placed close to the speaker, the effect of multipath distortion is limited in the signal used for QR code generation and can vary at the recording devices depending on the multipath acoustic environment.

(b) Environmental noise sources. Although any noise in the environment is recorded by both the embedding and recording devices, their intensities may vary due to the location of the noise sources. Typically, the impact of crowd noise, including indistinct voice or applause, is noticeable on the recording devices because of the close proximity but negligibly on the embedding device. Moreover, various other impulse and white noise sources are common in live speech settings that diversify the local effect of the undesired signals in the recording.

■ **Could we simply use speech-to-text conversion?**

Speech-to-text conversion can produce texts corresponding to the live speech which can potentially replace the features in the QR codes. While seems an appropriate choice, it fails in practice. The acoustic channel and background noise are different at the speaker and audience side which leads to a different interpretation of some words at the embedding and the verification causing false alarms. Moreover, Speech-to-text conversion does not capture the tone, pitch, or speed of the speech. Attackers can leverage this gap to create a caricature of the speech without changing the words as seen in an altered speech video of Nancy Pelosi that made her

[1] The Y axis shows up to 4kHz, since the normal human conversation in non-tonal languages like English is dominantly confined to this band. Figure 2 is taken from [56] with permission.

appear intoxicated during a press conference just by slowing down the speed of the speech [49].

3 SPEECH SIGNATURE EMBEDDING

3.1 Authentication Feature Generation

(a) Temporal Energy Modulation. The modulation of the sound intensity over time is considered the most salient acoustic feature in speech and it plays a crucial role in human perceptual speech comprehension as well as automatic speech recognition algorithms. Our first set of features (χ_{mod}) captures this phenomenon by averaging spectral energy over a short time window of the normalized spectrogram, as shown by the following equation. Here N is the number of frequency bins. Figure 3. a shows the capacity of this feature set in discriminating closely sounding words even when passed through a multipath channel.

$$\chi_{mod} = \frac{1}{N}\sum_{i=1}^{N} Y(f(i),t) \tag{1}$$

(b) Time-Frequency Convolutional Features *TalkLock* verification algorithm relies on the features to identify alteration of the speech and the features should be sensitive to even the slightest change in a spoken word. Sounds of two words can minimally differ by a phoneme which is the phonological unit that forms the fundamental set of possible sounds in a language. We design our features to be capable to distinguish such word couples, called minimal pairs. Figure 4 shows the spectrograms of a minimal pair ‘go’ and ‘no’. The temporal energy modulation feature alone cannot separate all minimal pairs as some phonemes can be manipulated to have the same average energy envelop over a time window differing from each other only by the formants or the concentration of acoustic energy at different frequencies. We develop a feature set (χ_{conv}) to capture the distribution of formant energy in the spectrogram using a two-dimensional convolution with a rectangular kernel $\mathcal{K}$. Figure 3.b shows the difference in Time-Frequency Convolutional features of two words recorded in a realistic multipath environment.

$$\chi_{conv} = flatten[conv(Y(f,t),\mathcal{K})] \tag{2}$$

3.2 Channel Normalization

To select features that can be matched regardless of the different channels and environmental noises. Figure 4 shows the spectrogram of two words. First, we apply a threshold-based filter on a spectrogram to remove all values less than some pre-defined threshold. This allows us to remove all time-frequency bins that have low amplitude because they are contributing very small information to the audio listened by humans. Then we observe the effect of the propagation channel on the recorded audio:

$$Y(f,t) = H(f,t) * S(f,t)$$

Here S is the actual speech, H is the channel, and Y is the speech recorded by the phone. We assume that for a short amount of time, the channel remains constant because both the speaker and recorder remain stationary.

$$Y(f,t) = H(f) * S(f,t)$$

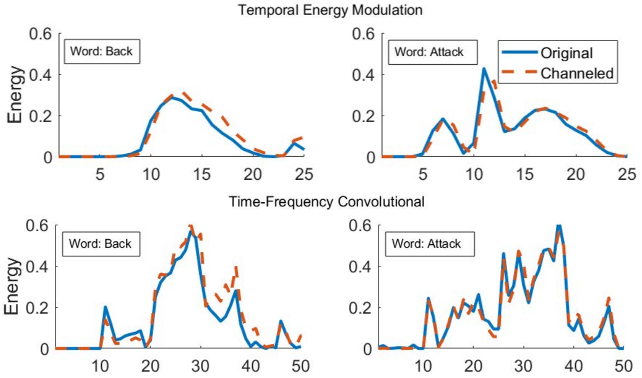

Figure 3: The comparison of (top-row) temporal energy modulation features, and (bottom-row) time-frequency convolutional features of two closely sounding words ‘back’ and ‘attack’ after passing through the channel.

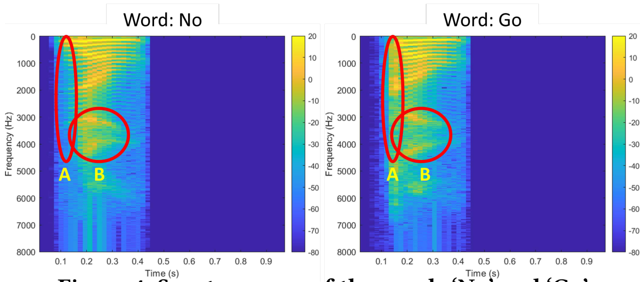

Figure 4: Spectrograms of the words ‘No’ and ‘Go’.

This equation tells us that for each frequency f, the channel is nothing but a multiplication with a constant complex number. So, if we normalize each frequency by the absolute maximum value for the same frequency over time, we can remove the effect of the channel. The step-by-step formulation is as follows: Compute the maximum absolute values for each frequency ‘f’. Let t_{max} be the time window for which speech S has the maximum absolute value

$$t_{max} = argmax(|S(f,t)|)$$

Since a channel is nothing but a constant multiplication, received signal Y also has the maximum absolute value at the same time window t_{max}

$$t_{max} = argmax(|Y(f,t)|) = |H(f)| * argmax(|S(f,t)|)$$

Finally, if we normalize the frequencies by their absolute maximum values, we can remove the effect of the channel.

$$\frac{|Y(f,t)|}{|Y(f,t_{max})|} = \frac{|H(f)| * |S(f,t)|}{|H(f)| * |S(f,t_{max})|} = \frac{|S(f,t)|}{|S(f,t_{max})|}$$

This allows us to match audio even when they are recorded at different locations, and by different devices.

3.3 Immutable Code Dissemination

Traditional meta-information-based authentication techniques assume access to the audiovisual content at the time during production and can embed the information directly into the content as metadata or obfuscated signals on the media. However, *TalkLock* attempts to protect live events and embed the information on the media without any control over the recording or content production process. An audience can record the live speech and the resulting content should have *TalkLock* meta-information embedded into the video data. To this end, it disseminates the meta-information in the form of a screen-camera communication between the live speech-signature embedding module used by the speaker and the camera that records the event. Next, we elaborate on the format for the meta-information followed by the QR-code-based communication concept.

(1) Meta-information format

The embedding module creates a packet of meta-information with features computed from the past two-time windows, packet index number, and a digital signature. Each time window provides 50 Temporal Energy Modulation feature values (χ_{mod}), and 100 Time-Frequency Convolutional feature values (χ_{conv}). Therefore, in total each packet contains 300 features. Figure 5 shows the arrangement and size of the data in the packet. The feature values are originally generated as single precision (16-bit) positive floating-point numbers less than one. We consider three decimal points of these values and packet them as short integers of size two bytes each. We use 256-bit Elliptical Curve private key encryption which generates a 64-byte digital signature. We combine 300 features (600 bytes), a packet index (2 bytes), and a digital signature (64 bytes) to generate a complete packet of size 666 bytes.

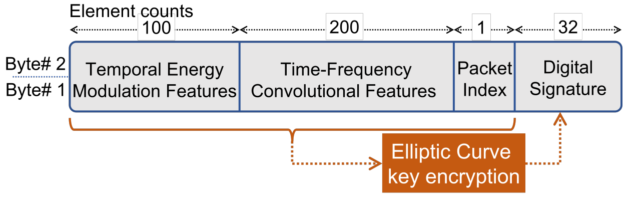
Element counts
100
200
1
32
Byte# 2
Byte# 1
Temporal Energy Modulation Features
Time-Frequency Convolutional Features
Packet Index
Digital Signature
Elliptic Curve key encryption

Figure 5: The meta-information packet format.

Packet index: We include an unsigned-integer index number to the meta-information packet index that wraps around after every 65536 iterations. This index number serves two purposes. First, as elaborated in the next section, it aids the packet decoding process to keep track of newly available information. Most importantly this index number helps the system to identify malicious editing where only frame sequences are altered without any changes to the content.

Daisy-chaining adjacent time windows: We embed meta-information of the previous two-time windows into each packet. This creates a daisy chain-like structure in which QR code number 'n' contains meta-information regarding time windows 'n-1' and 'n-2', as shown in Figure 6. The packet index number prevents an adversary from altering the frame sequences, but it cannot prevent a 'replay attack' from the same user's past content. Potentially an attacker can reuse a QR code with the same index number from past recordings. *TalkLock* stops such attack by daisy-chaining adjacent time-windows. When each window is encrypted, any change in the one-time window would require changing the entire stream of speech making opportunistic replacement practically impossible. Note here that the daisy-chain structure does not prevent the audience from recording a verifiable speech starting from an arbitrary point in time. Our verification process is independent of the starting and ending point of the recording and runs independently on each QR code. *TalkLock* starts the verification process after two-time windows of speech duration have passed and the first QR code is recorded by the audience till the end of the recorded speech.

Digital signature: We anticipate that an equipped adversary will attempt to avoid detection by carefully manipulating the meta-information to match the altered speech. We use a cryptographic digital signature to prevent tampering with the data in the packet. The system uses a 256-bit private key of the speaker to generate the 64-byte signature over the entire byte array containing features and packet index sign, as shown in Figure 5. This private key is an information secret to the adversary and therefore cannot successfully generate a new or modify an existing meta-information packet without getting detected by the signature authentication process. The system applies Elliptic Curve key encryption for the signature, which is a popular and tested method adopted by large-scale messaging applications including, WhatsApp [2], Facebook [5], and Signal [4].

(2) Screen-camera communication with QR codes

A meta-information packet is generated for each time window of the speech signal. We use a QR-code-based screen-camera communication to disseminate the meta-information so that it will automatically get embedded in the content during recording. QR codes are originally designed for short range (less than 30 cm) and small data exchange to camera-enabled mobile devices [24, 29, 32, 40]. Later QR code format and decoding process evolved to communicate at a high data rate of several Mbps and over a large distance of tens of meters [53]. We select QR code-based embedding for its robustness to variations of commodity cameras, recording angles, and surrounding environments.

To date, there are 40 versions of QR codes with different data capacities and ranges of operation. We use QR code version-18 with a data capacity of 741 bytes for the meta-information packet. The system generates a QR code from the packet in each time window, as shown in Figure 6. A QR code is displayed on the screen of the speaker's live speech-signature embedding module. We limit the size of the QR code maximum dimension to 3 inches that fit on average smartphone screens. Our experiments show that the recording device can capture and successfully decode the QR codes from more than 20 feet. Naturally bigger QR codes displayed on a tablet, or an LCD screen can further increase the range of decoding for specific application scenarios.

4 SPEECH INTEGRITY VERIFICATION

The verification process first separates the speech signal from the audiovisual content under test and focuses on two-time windows of speech at a time. This section explains the steps involved in this integrity verification process.

4.1 Speech Denoiser

For integrity verification, *TalkLock* compares the features computed from the speech signal under test with the features embedded into the QR code. However, it is not feasible to compare these features when there is a strong local noise present at the receiver

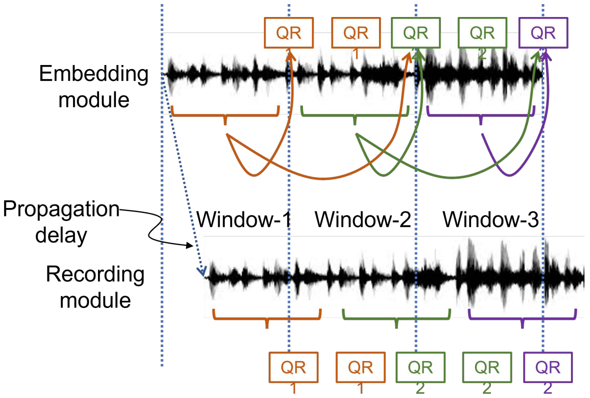

Figure 6: The timings of the QR code generation and time-windows.

end, such as phone ringing, clapping, or babbling noise. We add a speech-denoising module to eliminate such spurious sounds. We have adopted a state-of-the-art speech enhancement model proposed in [20] to implement the denoiser. The model is based on an encoder-decoder architecture optimized on both time and frequency domains using multiple losses.

4.2 Feature Recovery from QR Sequence

Each frame of the video contains a packet index number in a QR code that changes at the beginning of each time window and holds the meta-information packet corresponding to the previous two windows. The system first decodes the QR code using standard libraries and recovers the meta-information packet from the code. It then decrypts the signature from the packet using the speaker’s public key and compares it with the hash of the data part of the packet. In case of a failed match, the system considers the QR code for the current time window to be compromised and moves on to the next window after raising an alert. In case of successful authentication, the system proceeds to packet index verification. Given the QR codes are displayed for a whole-time window, multiple video frames may capture the same QR code. The system waits for an unseen packet index number to proceed with discarding the duplicates. However, an out-of-sequence packet index indicates a malicious frame swap or deletion. The system raises an alert and continues processing for the next time window. After passing the signature and packet index verification, the feature vectors ($\chi_{QR} = [\chi_{mod}, \chi_{conv}]$) are recovered from the packet.

The next step is to generate features (χ_{speech}) from the previous two-time windows of speech separated for verification against the features recovered from the QR code. However, this step requires precisely identifying the time window boundaries from the received speech signal or the features may not match with that from the QR code. The use of the wrong audio chunk for feature extraction is a crucial problem because the spectrogram varies quickly with time. Note that the timing of the windows is defined by the embedding module at the speaker side and there are no explicit information exchange embedding and verification modules. The QR codes generated at the edge of the time windows could be an indicator of window timings. However, these timings are imprecise as there can be a maximum gap of one video frame between the QR code displayed and the recorder first capturing a complete view of it. Moreover, the sound takes some time to travel from the speaker to the recording device, called signal propagation delay, but the QR code is seen by it almost instantly leading to a mismatch between the QR code and speech window timings. We elaborate to mitigate this synchronization issue.

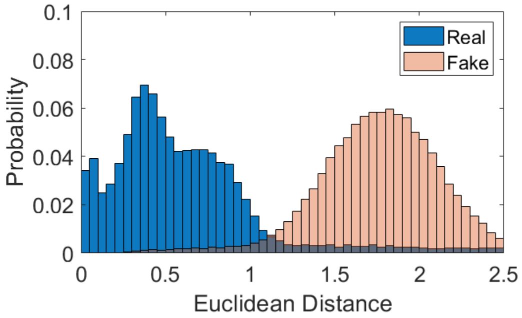

Figure 7: Histograms of Euclidean distances between original features with that of the real and modified speeches.

4.3 Frame Synchronization

We target to deal with the window synchronization challenge using a correlation method. We use the new QR codes as a coarse timing for the windows. Next, instead of picking only the previous time window for processing, we pick a longer duration of the speech signal in a way that it encompasses the audio received after a propagation delay. Then use a one-time window of data and slide the window over the speech signal in iterations. For each window, the system computes the feature set and correlates these features with the features extracted from the QR code. The maximum correlation happens when the sliding window matches with a majority, if not all, of the features. We define this maximum correlation point as the time-window boundary. Note that once a boundary is found, it can be used to find the subsequent windows.

After window synchronization, the next step is to generate features (χ_{speech}) from the previous time window of the speech applying the same feature generation algorithm used in the embedding process (Section 3). These two sets of features, χ_{QR} and χ_{speech}, are saved for verification.

4.4 Channel-agnostic Feature Verification

The final step compares the feature vectors from the speech and the QR codes to verify authenticity. However, due to environmental noise and multipath distortion, these two feature sets do not exactly match even legitimate speech recordings. One approach to allow small differences in the features is to use the Euclidean distance or similarity score. Figure 7 shows a clear difference between the distances with real and modified audio. But for the final classification, we need to define a threshold. A pre-defined threshold may work for a certain acoustic environment and noise scenario but is likely to perform inadequately in a different environment. We used a deep learning-based approach instead.

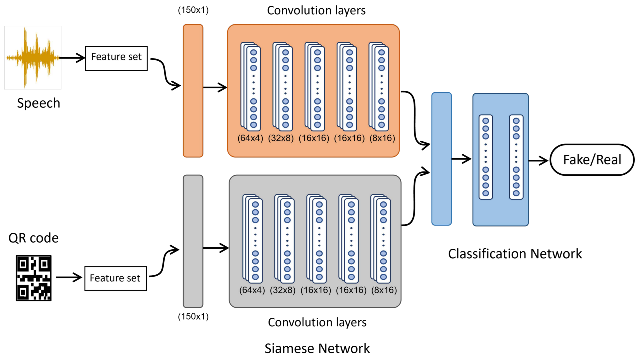

Figure 8: Network design of a proposed classifier.

We refer to deep learning which shows amazing performance in image classification [35, 45] and signature verification [21, 27] which are close to feature verification with minute variations. From the collection of deep learning models for signature verification, we choose Siamese networks that contain two similar networks that run in parallel with weights shared between them. Both original features (i.e., decoded from QR code) and the features under verification (i.e., extracted from audio) are passed through the network which projects these features into a different domain. These networks are trained in a manner so that it projects features of authentic audio (even with channel variation and noise) closer to the true features, and features of modified audio far from the true features. Once the projection is done, it needs a classifier for the final authenticity result. For this, we train a 4-layer neural network-based classifier for final real/fake classification. Figure 8 shows the architecture of our both Siamese network and the neural classifier. The Siamese network comprises 5 1-Dimensional convolution layers with ReLU activation containing 4, 8, 16, 16, and 16 filters respectively.

Arguably, any deep-learning model can be vulnerable to carefully crafted adversarial attacks. However, adversarial attempts often leverage the full or partial knowledge of the model (white-box attack) [15, 16, 69], and keeping the model parameters secret can prevent such attacks. On the other hand, a black-box attack (i.e., without the knowledge of the model) is still possible, as shown by some recent papers [17, 23], if the adversary can observe the model's response to enough number of queries. In *TalkLock*, the system can keep the model parameters hidden and limit the number of queries allowed for a given audiovisual content per user to avoid such adversarial attacks.

4.5 Obtaining the Public Key

The public key of the speaker is required to verify the digital signature in the meta-information packet. The features signed by the speaker's private key can only be verified by the corresponding public key. That is why we are signing before embedding selected features. Otherwise, anyone can fool our system by modifying both audio and embedded features. Therefore, we need to devise a method in which the verifier can ensure that the public key corresponds to the speaker in the video. There are two approaches to obtaining a public key. First is by using Transport Layer Security (TLS) certificates, which are primarily used to open an encrypted channel with the web server, encrypt email and voice over IP [18]. Certificate authorities, after verification issues a signed digital certificate containing the sender's public key. TLS certificates allow anyone to verify the identity of the sender by verifying the authenticity of a digital certificate using publicly available keys of certificate authorities [10]. The limitation of this method is that anyone who wants to verify a video needs to contact a speaker to obtain a public key. The second is by maintaining an app database server that contains a list of public keys corresponding to each user. Similar to Facebook, and WhatsApp encryption protocols, upon installation, the app generates a key pair and shares its public key with the app server. Note that keys are generated inside the speaker's device so the private key has never passed through the network and the server has no information about private keys.

4.6 Dynamic QR Code Refresh Rate

TalkLock disseminates a meta-information packet by embedding a complete packet into a QR code. Since the packet comprises 666 Bytes, we need to use a QR code of at least version 18. The complexity of a QR code increases with the version of the QR code, which can decrease the decoding robustness of the QR code in live speeches. So, rather than using a single complex QR code to embed a complete packet, we aim to use a smaller/less complicated version of the QR code. We achieve this by dividing a packet into 'n' smaller chunks. Then we increase the refresh rate of the QR code accordingly by 'n' times because we still want to disseminate a complete packet of 666 Bytes. At the verification end, the system can determine the factor 'n' from the version of the QR code. The system extracts data from 'n' consecutive QR codes and combines them to form a complete packet.

5 EVALUATION

This section initially discusses the implementation details and experimental setup. Then it evaluates the performance of *TalkLock*, under various practical scenarios.

5.1 Experimental Setup

Setup: We develop a prototype for *TalkLock* using Python and Matlab. To collect diverse scenarios of data, we emulate a live speaker using a laptop speaker. The live speech-signature embedding module runs in real-time on a laptop. We record the audio and video of the person using an iPhone 11 and Canon EOS camera with a QR code visible in the captured frame. We use open-source libraries, QR code [7] and pyzbar [6], to generate and decode QR codes. To apply the cryptographic signatures, we use the Elliptical Curve Digital Signature Algorithm (ECDSA). We record and store the video to process it offline and verify the integrity of the content. Figure 10 shows an experimental setup of *TalkLock*.

Training Dataset: To train the Siamese network, we create our training dataset from the LibriSpeech ASR corpus [52], which contains spoken sentences from 40 different persons. We randomly choose 200 seconds of speech data for 30 speakers and convolve it

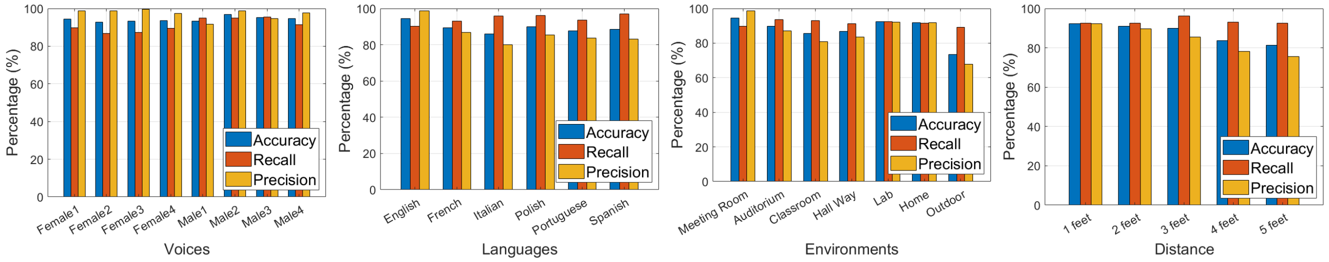

Figure 9: Classification performance under different a) voices, b) languages, c) environments, d) distance between speaker and a QR code generator.

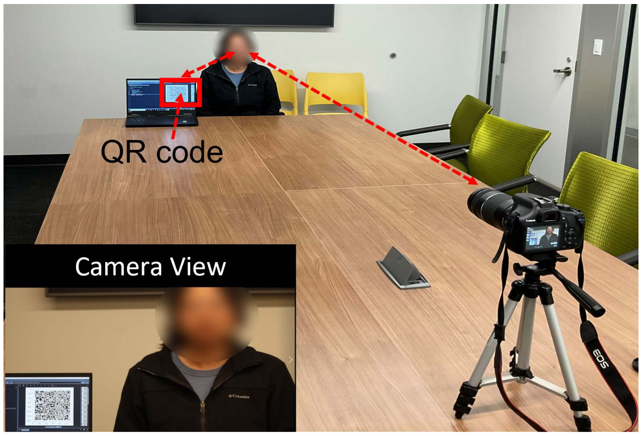

Figure 10: Experimental Setup with the human speaker, camera location, and the speech signature embedding module implemented on a laptop.

with 8 unique room impulse responses [61] to add the variation of different acoustic environments. We generate synthetically altered recordings, or negative examples, with the same set of speakers and room impulse responses. Finally, we get 800 minutes of original and altered data points each for our training dataset.

Performance Metrics: *TalkLock* predicts the authenticity of the audio-visual media content. This is a single class classification problem with the output either *real* or *fake*. To evaluate the performance of this system, we used classification accuracy, recall, and precision as our metrics. We also calculate the root mean squared error (RMSE) between the authentic features and extracted features from the recorded audio. A low RMSE score signifies that the extracted features are similar to the authentic features which are used to create the QR code.

5.2 End-to-End Performance

This section summarizes the overall performance of *TalkLock*. We evaluated our system under eight different voice types, six languages, eight types of noise sources, and seven types of acoustic environments. We also performed experiments with varying distances between the speaker and QR code-generating module from 1 foot to 5 feet. The overall results show that *TalkLock* achieves 87.59%, 90.52%, and 87.15% of mean accuracy, recall, and precision respectively. Next, we will discuss and analyze the evaluations in detail.

5.3 Sensitivity Analysis

Impact of different voices: Ideally, a verification system should work for any voice without any additional calibration or training. We test *TalkLock*'s performance with eight different voices (four male and four female) and the 10,000 most common words taken from Google's Trillion Word Corpus. Note that these voices are completely different from the voices present in our training dataset. We place the laptop running live speech-signature module one foot from the speaker and the recording camera three feet from the speaker. We captured 80 minutes of speech data, comprising approximately 4800 words, containing an equal amount of original and synthetically altered recordings. To obtain the original speech data, we recorded a 5-minute live speech session for each speaker while simultaneously running a signature embedding module for QR codes. Next, to generate the altered speech data, we recorded an additional 5-minute session for each speaker, with different content from the original speech. We then replaced the audio of the authentic speech with new audio to produce a synthetically altered speech. The purpose of creating a set of altered recordings with the same voice but different words is to simulate an adversary who may try to modify the content by changing the words while preserving the original speaker's voice. Figure 9.a shows that *TalkLock* achieves 94%, 91%, and 97% of mean accuracy, recall, and precision respectively.

Impact of different languages: We evaluate the system using the Multilingual LibriSpeech dataset [54] which contains spoken sentences in six different languages English, French, Italian, Polish, Portuguese, and Spanish. Figure 9.b shows that *TalkLock* achieves more than 90% accuracy on all languages and proves to be agnostic to languages.

Impact of acoustic environments: We performed experiments in seven different acoustic environments: meeting room, auditorium, classroom, hallway, lab, home, and outdoor. Each acoustic environment has a unique multi-path profile and ambient noise. Figure 11 shows our setup in few of these locations. In Figure 9.c, we see that *TalkLock* achieves a mean classification accuracy of 87.7%. The histogram shows that the performance in the outdoor environment is less than that of the indoor environment. This is due to the presence of higher ambient noise from passing traffic and pedestrians.

Impact of environmental noise: We tested *TalkLock* in varying types of noise conditions in both simulation and real-world experiments. For simulation data, we create the data by applying

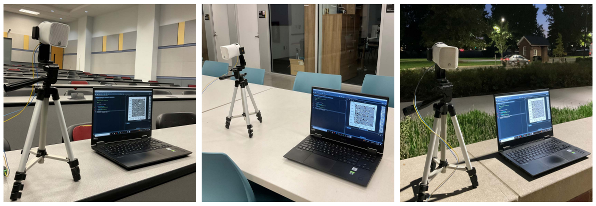

Figure 11: System evaluation at various locations a) Auditorium, b) Hallway, c) Outdoor.

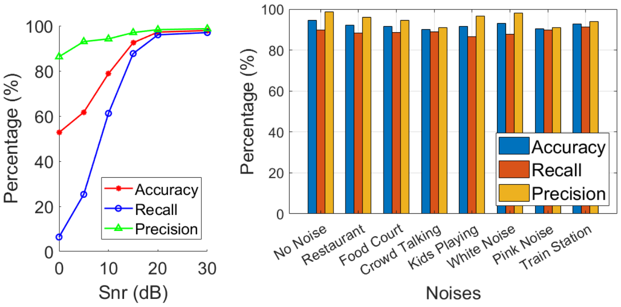

Figure 12: Classification accuracy under different a) SNR levels, b) types of noises.

room impulse responses (RIR) from the MIT RIR dataset [61]. Then, we add different levels of Additive White Gaussian Noise. Figure 12.a shows that with 0 dB SNR, *TalkLock*'s classification accuracy is close to 50% which is no better than a random guess in two-class classification. The accuracy increases with the increase in SNR level and reaches above 90% after 20 dB of SNR level. For real-world experiments, we play various noises in the background using a digital audio speaker. The SNR level is maintained at 30dB at the laptop, which is generating the QR code, and 20dB at the recording smartphone. From experimental results shown in Figure 12.b, we see that with varying noise levels *TalkLock* achieves 90% classification accuracy.

Effect of distance: We evaluate the performance of our system with varying distances between the speaker and QR generator from 1 foot to 5 feet. Other parameters like voice type, loudness, and location were fixed. Figure 9.d shows that within 2 feet, the classification accuracy is above 90%. We observe that the performance degrades with the distance because of the reduction in SNR of sound with distance.

Live speech experiment: In this section, we evaluate our system on live speech. For this experiment, we recorded live speeches of 4 different speakers. Our study has been approved by the IRB of our institution. For each user, we collected two separate speeches of 60 seconds each. We recorded the speech video using an iPhone 11. These are authentic speech examples that we want *TalkLock* to classify as real. To create doctored footage, we swap the audio of two speeches from the same speaker. This represents the best possible speech modification attack because it has no artifacts and anomalies which are generally present in fake content. We collected 480 seconds of authentic and 480 seconds of fake speech to *TalkLock* for verification. Figure 13 shows that our system is able to accurately identify real and fake content with 85% accuracy.

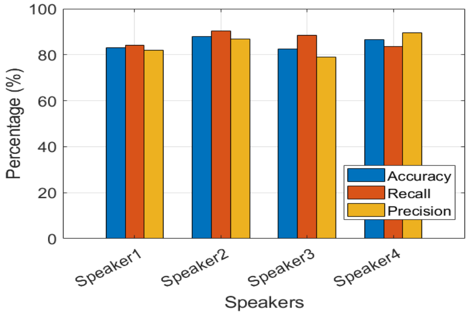

Figure 13: Classification performance of *TalkLock* under live speech recordings.

Computational complexity: In this section, we analyze the computational complexity of *TalkLock*. The signature embedding module has a constant time complexity of O(1) as it performs the same commands to compute features and generate QR codes for each time window. On the other hand, the verification module has a linear time complexity of O(n), where 'n' is the number of time windows in the recorded speech, as the verification process is independent for each time window. We implemented the prototype on a laptop with an Intel i7 10th generation processor, 32 GB RAM, and Nvidia GeForce 2060 GPU, and evaluated the execution time for both modules. The signature embedding module takes 0.157 seconds, which is less than the duration of the time window (1 second) necessary to ensure real-time dissemination of the meta-information. The verification module takes 1.223 seconds to verify each time window, which is acceptable as verification can run offline. Note that the execution time of both modules depends on the device's CPU, RAM, and GPU specifications.

5.4 Micro-benchmarks

In this section, we perform micro-evaluations to understand the effect of system design parameters and the robustness of our system.

Effect of channel normalization and frame synchronization: To highlight the importance of channel normalization and frame synchronization, we performed a benchmarking experiment. We picked a sample feature embedded into the QR code and a feature computed from the authentic recorded audio. Figure 15.a shows that after applying both channel normalization and frame synchronization, features embedded into the QR code and features computed from the recorded audio follow a similar trend. It also shows that without frame synchronization and channel normalization computed features are not matching with the features embedded in the QR code. This highlights the importance of channel normalization and frame synchronization in matching features from the recorded audio. Figure 15.b shows a cdf plot comparing the RMSE between features with and without using Channel normalization and frame synchronization.

Verification is robust to online media formats: In this section, we experimented to evaluate the robustness of *TalkLock* under diverse devices and social media sites. Different devices and social media sites often use different video and audio codecs. Codec is a program that encodes and decodes data streams. We tested

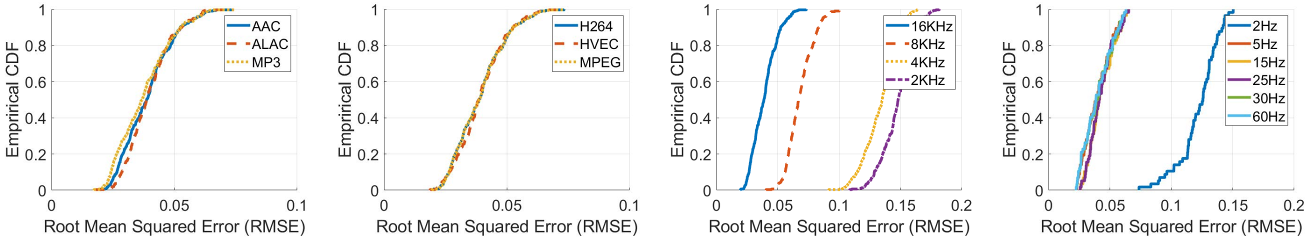

Figure 14: RMSE between features a) under various audio formats, b) video formats, c) audio sampling rate, and d) video frame rate. (Note: the x-axis for figures 1 and 2 is from 0 to 0.1, and for figures 3 and 4 is from 0 to 0.2).

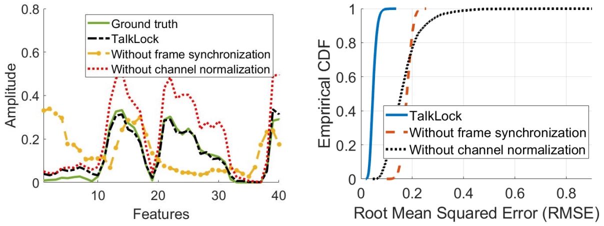

Figure 15: a) Comparing embedded and recorded features with and without applying channel normalization and frame synchronization, and b) cdf of their RMSE.

our system under the following three most commonly used audio codecs: Advanced Audio Codec (AAC) [22], Apple Lossless Audio Codec (ALAC), and MP3 [59]. We also tested our system under the three most common video codecs: H.264 [66], HVEC, and MPEG [38]. Figure 14.a shows that the cdf plot of RMSE between embedded features and features computed from the recorded audio does not change with different audio formats. Similarly, Figure 14.b shows that features are robust to various video formats. We also evaluate the performance for different sampling and frame rates. Figure 14.c shows the cdf of RMSE between features using different sampling rates. This plot highlights that error drastically increases when the sampling rate is reduced to 4KHz which is reasonable because human speech has meaningful information till 4KHz and by sampling at 4KHz we are getting rid of useful information. Figure 14.d shows the cdf of RMSE between features using different frame rates. This shows that as long as the frame rate is greater than 5Hz, features remain robust to variations in frame rate.

Effectiveness of features:
TalkLock is using temporal energy modulation and time-frequency convolutional features. We choose these two features because of their ability in detecting a minute difference in sounds which makes them capable of detecting a difference in closely sounding words called word families. For example, attack, back, and knack correspond to the same word family. To evaluate how our system performs when a word is replaced with a closely-sounding word, we simulated the following experiment. We collected a list of 50 different word families, and each family contains 10 closely-sounding words. For each word family, we compare all words with each other. To test, if classification accuracy is robust to practical scenarios, we performed the above-mentioned experiment by applying 30 different channel responses from the MIT channel responses data set. Figure 16 shows the cdf of classification accuracy. It shows that our system achieves a median of 85% accuracy in this highly sophisticated attack. To test the effectiveness of the designed features, we performed the same experiment by using only one kind of feature at a time. Figure 16 shows that median classification accuracy decreases to 77% and 68% by individually using temporal energy modulation and time-frequency convolutional features respectively.

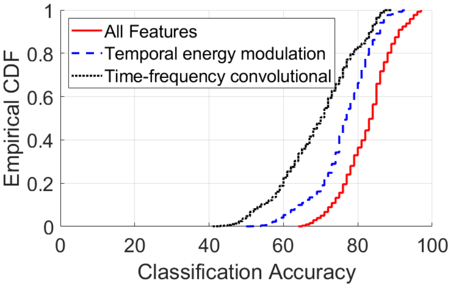

Figure 16: Classification accuracy with different numbers of features.

6 LIMITATIONS AND FUTURE WORK

In this section, we will discuss the system's current behavior, expected outcomes, and potential future directions.

Impact of ambient noise: *TalkLock* is designed to capture any malicious modification to the speech. It may encounter challenges when high levels of ambient noise are present only near the recording device, such as crowd talking, audience clapping, phone ringing, etc. The addition of a local noise near the recording device will cause our system to flag authentic speech as tampered. The reason for this misclassification is that speaker's microphone may not capture these local noises, which can result in a mismatch between the meta-information embedded in the QR code and features computed from the recorded speech. While our system can effectively ignore low ambient noise added to the speech with the help of a speech denoiser, further processing is required to remove any non-speech component from the recorded audio to obtain a clean speech before running the verification process. It is reasonable to remove ambient noise because the goal of our system is to identify any contextual speech modifications. This will enable the audience to record a verifiable speech even in a noisy crowd.

Missing QR codes: The current implementation of *TalkLock* relies on the screen-camera communication to transmit QR codes required for speech verification. However, a missing QR code leaves content ineligible for verification, which adds a constraint on the audience to always keep the QR code in the recorder's field of view. To address this issue, *TalkLock* proposes to display a QR code on a smartphone placed near the speaker's face which makes it easier for the audience to record both speaker and the QR code. However, this solution is not effective in scenarios where either the audience or the speaker is in motion, the camera is focused solely on the speaker's face, the recorder is too far to capture a clear QR code, or the camera resolution is insufficient to record a readable QR code. To overcome this limitation, it is necessary to explore other channels of transmitting meta-information to the live audience. In our future work, we plan to investigate the use of acoustic, WiFi, and Bluetooth-based communication channels for meta-information broadcasting. The key challenge is to ensure that the communication channel will open ubiquitously without imposing any additional requirements on the audience.

Securing visual features: Currently, *TalkLock* is effective in protecting speech audio from contextual modifications. However, it does not consider the rich contextual information present in speech videos. Adversaries can modify the facial expressions, hand gestures, and body postures of the speaker, which can completely alter the perceived meaning of the speech, e.g., a wink or a thumb down. As our future direction, we aim to design robust visual features that can be embedded into the QR code to enable our system to verify both the audio/visual content of the recorded.

7 RELATED WORK

This section discusses the existing methods of detecting and preventing content modification attacks.

7.1 Artifact-based Detection

The creation and detection of fake media is an active field of research [46, 50]. Existing methods rely on subtle inconsistencies, such as warping [42], blending [39], and splicing [33] added into the content either during the generation or modification process. The signatures corresponding to the common content generators are also used in detecting fake media [65, 68, 71, 72]. The detection is not only limited to capturing technical inconsistencies, subtle abnormalities in human behaviors can also be used to verify content integrity. Such as, unnatural eye blinking [41], inconsistent head poses [67], abnormal heart rate [31], mismatch between lip movement and spoken words [12, 28] and emotional cues from both audio and video [47]. However, these detection techniques rely on inconsistencies and signatures that can be learned and resolved by new fake content generation methods, so it is necessary to continue developing new detection techniques. In *TalkLock*, we take a different approach and propose a detection method that verifies the content by matching features of the speech signal with the authentic features from the QR code embedded into the video.

7.2 Metadata-based Prevention

Another approach to mitigating fake content is by embedding immutable meta-information which can be used to verify the content's authenticity. Content Authentication Initiative (CAI) has developed a way of appending critical information into the metadata of the image [3]. Recently, Vronicle [44] propose a system for generating verifiable videos by appending recorder credentials and information about the applied filters in the metadata. On the other hand, [30] proposes a deep learning-based image authentication method that uses a watermark embedded in the image in verifying its authenticity. Another work [43] proposes an image authentication method utilizing a QR code as a watermark, which contains a cryptographically secure digital signature of the image. Similarly, [11] proposes a detection method for voice impersonation by adding digital watermarks in the audio track of the video. These approaches need the recorder's cooperation to add verification information.

We recently found a preprint of a short note discussing a similar idea of QR code-based screen-camera communication for live speech authentication [19]. This work used a textual representation of the speech, as opposed to the voice signal used in *TalkLock*, as the source information to protect from the adversary. While innovative, the textual representation of speech is inadequate to protect against the most common attacks through the manipulation of tone, pitch, and speed of the speech, as seen in the case of the fake Nancy Pelosi video [49]. *TalkLock* presents a technique to identify the set of signal-level features that can capture the phonetic information of the speech as it is heard by the audience. The features are carefully designed to remain robust against environmental channels and form an immutable code for the speech. Moreover, we establish a QR code-based communication system and packet structure to convey the meta-information allowing the audience to record a verifiable video. *TalkLock* designs and implements a practical system and shows a comprehensive evaluation of a technique to protect live speeches in real-world scenarios.

7.3 Regulation at Source

Another way of controlling the spread of fake media is by taking regulatory measures and increasing media literacy. In 2019, Texas passed a law making it illegal to distribute deepfake videos [9] that can injure a candidate or influence the election results. Moreover, in 2018, Nebraska Senator Ben Sasse introduced the malicious deepfake prohibition act which makes it illegal to distribute a deepfake that can facilitate criminals [1]. Several initiatives are in the proposal phase such as the deepfakes accountability act requires that malicious audio and visual content should be marked as deepfakes. In addition to regulatory measures, raising media literacy and teaching audiences to be curious about the sources of information and assess their credibility is another way to reduce the impact of fake media [8]. However, *TalkLock* takes a different approach and attempts to provide a technical solution for content authentication.

8 CONCLUSION

In this paper, we are presenting *TalkLock*, a speech integrity verification system that enables speakers to deliver tamper-proof live speeches even when the speech is recorded by any member of the audience. *TalkLock* achieves this by proposing a novel verification system in which meta-information is added to the video by displaying it on a QR code. Moreover, it designs two types of features and a Siamese network for channel-agnostic verification.

REFERENCES

[1] https://www.congress.gov/bill/116th-congress/house-bill/3230. Last accessed 09 December 2022.

[2] About end-to-end encryption. https://faq.whatsapp.com/820124435853543. Last accessed 07 April 2023.

[3] Content authenticity initiative. https://contentauthenticity.org/. Last accessed 09 December 2022.

[4] Documentation. https://signal.org/docs/. Last accessed 07 April 2023.

[5] messenger secret conversations technical whitepaper. https://about.fb.com/wp-content/uploads/2016/07/messenger-secret-conversations-technical-whitepaper.pdf. Last accessed 07 April 2023.

[6] python one dimensional barcode and qr code reader. https://pypi.org/project/pyzbar/. Last accessed 09 December 2022.

[7] python qr code generator. https://pypi.org/project/qrcode/. Last accessed 09 December 2022.

[8] Securing american elections. https://docslib.org/doc/5559699/securing-american-elections. Last accessed 09 December 2022.

[9] Texas sb751: 2019-2020: 86th legislature. https://legiscan.com/TX/text/SB751/id/2027638. Last accessed 09 December 2022.

[10] Digital certificates and certificate authorities. https://www.ibm.com/docs/en/db2/11.1?topic=SSEPGG_11.1.0%2Fcom.ibm.db2.luw.admin.sec.doc%2Fdoc%2Fc0053515.html, Apr 2021. Last accessed 09 December 2022.

[11] Sakshi Agarwal and Lav R Varshney. Limits of deepfake detection: A robust estimation viewpoint. *arXiv preprint arXiv:1905.03493*, 2019.

[12] Shruti Agarwal, Hany Farid, Ohad Fried, and Maneesh Agrawala. Detecting deep-fake videos from phoneme-viseme mismatches. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 660–661, 2020.

[13] Yang Bai, Nakul Garg, and Nirupam Roy. Spidr: Ultra-low-power acoustic spatial sensing for micro-robot navigation. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, pages 99–113, 2022.

[14] Yang Bai, Nakul Garg, Harshvardhan Takawale, Anupam Das, and Nirupam Roy. Natural voice interface for the next generation of smart spaces. In *Proceedings of the 24th International Workshop on Mobile Computing Systems and Applications*, pages 140–140, 2023.

[15] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David A Wagner, and Wenchao Zhou. Hidden voice commands. In *Usenix security symposium*, pages 513–530, 2016.

[16] Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE security and privacy workshops (SPW)*, pages 1–7. IEEE, 2018.

[17] Yuxuan Chen, Xuejing Yuan, Jiangshan Zhang, Yue Zhao, Shengzhi Zhang, Kai Chen, and XiaoFeng Wang. Devil's whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices. In *USENIX Security Symposium*, pages 2667–2684, 2020.

[18] Cloudflare. What is transport layer security? https://www.cloudflare.com/learning/ssl/transport-layer-security-tls/. Last accessed 09 December 2022.

[19] Andrew Critch. Wordsig: Qr streams enabling platform-independent self-identification that's impossible to deepfake. *arXiv preprint arXiv:2207.10806*, 2022.

[20] Alexandre Defossez, Gabriel Synnaeve, and Yossi Adi. Real time speech enhancement in the waveform domain. *arXiv preprint arXiv:2006.12847*, 2020.

[21] Sounak Dey, Anjan Dutta, J Ignacio Toledo, Suman K Ghosh, Josep Lladós, and Umapada Pal. Signet: Convolutional siamese network for writer independent offline signature verification. *arXiv preprint arXiv:1707.02131*, 2017.

[22] Bluetooth Doc. Advance audio distribution profile specification. *Adopted version*, 1.

[23] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7714–7722, 2019.

[24] Wan Du, Jansen Christian Liando, and Mo Li. Softlight: Adaptive visible light communication over screen-camera links. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE, 2016.

[25] Kaylee Fagan. A viral video that appeared to show obama calling trump a 'dips—' shows a disturbing new trend called 'deepfakes'.

[26] Nakul Garg, Yang Bai, and Nirupam Roy. Owlet: Enabling spatial information in ubiquitous acoustic devices. In *The 19th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*. ACM, 2021.

[27] Luiz G Hafemann, Robert Sabourin, and Luiz S Oliveira. Learning features for offline handwritten signature verification using deep convolutional neural networks. *Pattern Recognition*, 70:163–176, 2017.

[28] Alexandros Haliassos, Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Lips don't lie: A generalisable and robust approach to face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5039–5049, 2021.

[29] Tian Hao, Ruogu Zhou, and Guoliang Xing. Cobra: Color barcode streaming for smartphone systems. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 85–98, 2012.

[30] Todd C Helmus. Artificial intelligence, deepfakes, and disinformation: A primer. Technical report, RAND CORP SANTA MONICA CA, 2022.

[31] Javier Hernandez-Ortega, Ruben Tolosana, Julian Fierrez, and Aythami Morales. Deepfakeson-phys: Deepfakes detection based on heart rate estimation. *arXiv preprint arXiv:2010.00400*, 2020.

[32] Pengfei Hu, Parth H Pathak, Xiaotao Feng, Hao Fu, and Prasant Mohapatra. Colorbars: Increasing data rate of led-to-camera communication using color shift keying. In *proceedings of the 11th ACM conference on Emerging Networking experiments and technologies*, pages 1–13, 2015.

[33] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A Efros. Fighting fake news: Image splice detection via learned self-consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 101–117, 2018.

[34] Tim Hwang. Deepfakes: A grounded threat assessment. *Centre for Security and Emerging Technologies, Georgetown University*, 2020.

[35] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, page 0. Lille, 2015.

[36] Marcus Krieg. Why video is the most persuasive form of content.

[37] Oxana Lapteva. *Speaker Perception and Recognition: An Integrative Framework for Computational Speech Processing*. kassel university press GmbH, 2011.

[38] Didier Le Gall. Mpeg: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4):46–58, 1991.

[39] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5001–5010, 2020.

[40] Tianxing Li, Chuankai An, Xinran Xiao, Andrew T Campbell, and Xia Zhou. Real-time screen-camera communication behind any scene. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pages 197–211, 2015.

[41] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *2018 IEEE International workshop on information forensics and security (WIFS)*, pages 1–7. IEEE, 2018.

[42] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. *arXiv preprint arXiv:1811.00656*, 2018.

[43] Xiaomei Liu and Xin Tang. Image authentication using qr code watermarking approach based on image segmentation. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1572–1577. IEEE, 2020.

[44] Yuxin (Myles) Liu, Yoshimichi Nakatsuka, Ardalan Amiri Sani, Sharad Agarwal, and Gene Tsudik. Vronicle: Verifiable provenance for videos from mobile devices. MobiSys '22, page 196–208, New York, NY, USA, 2022. Association for Computing Machinery.

[45] Iaroslav Melekhov, Juho Kannala, and Esa Rahtu. Siamese network features for image matching. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 378–383. IEEE, 2016.

[46] Yisroel Mirsky and Wenke Lee. The creation and detection of deepfakes: A survey. *ACM Computing Surveys (CSUR)*, 54(1):1–41, 2021.

[47] Trisha Mittal, Uttaran Bhattacharya, Rohan Chandra, Aniket Bera, and Dinesh Manocha. Emotions don't lie: An audio-visual deepfake detection method using affective cues. In *Proceedings of the 28th ACM international conference on multimedia*, pages 2823–2832, 2020.

[48] Paarth Neekhara, Shehzeen Hussain, Xinqiao Zhang, Ke Huang, Julian McAuley, and Farinaz Koushanfar. Facesigns: Semi-fragile neural watermarks for media authentication and countering deepfakes. *arXiv preprint arXiv:2204.01960*, 2022.

[49] AP news. Altered video makes pelosi seem to slur words. https://apnews.com/article/social-media-donald-trump-nancy-pelosi-ap-top-news-not-real-news-4841d0ebcc704524a38b1c8e213764d0. Last accessed 09 December 2022.

[50] Thanh Thi Nguyen, Quoc Viet Hung Nguyen, Cuong M Nguyen, Dung Nguyen, Duc Thanh Nguyen, and Saeid Nahavandi. Deep learning for deepfakes creation and detection: A survey. *arXiv preprint arXiv:1909.11573*, 2019.

[51] Ewan Palmer. Trump deepfake shows ex-president joining russia's version of youtube, Mar 2022.

[52] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

[53] Samuel David Perli, Nabeel Ahmed, and Dina Katabi. Pixnet: Interference-free wireless links using lcd-camera pairs. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 137–148, 2010.

[54] Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. Mls: A large-scale multilingual dataset for speech research. *ArXiv*, abs/2012.03411, 2020.

[55] Amna Qureshi, David Megías, and Minoru Kuribayashi. Detecting deepfake videos using digital watermarking. In *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1786–1793. IEEE, 2021.

[56] Nirupam Roy and Romit Roy Choudhury. Listening through a vibration motor. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, pages 57–69. ACM, 2016.

[57] Irtaza Shahid, Yang Bai, Nakul Garg, and Nirupam Roy. Voicefind: Noise-resilient speech recovery in commodity headphones. In *Proceedings of the 1st ACM International Workshop on Intelligent Acoustic Systems and Applications*, pages 13–18, 2022.

[58] Tom Simonite. A zelensky deepfake was quickly defeated. the next one might not be, Mar 2022.

[59] Jonathan Sterne. *MP3: The meaning of a format*. Duke University Press, 2012.

[60] Ingo R Titze. *Principles of voice production*. National Center for Voice and Speech, 2000.

[61] James Traer and Josh H McDermott. Statistics of natural reverberation enable perceptual separation of sound and space. *Proceedings of the National Academy of Sciences*, 113(48):E7856–E7865, 2016.

[62] Daniel Victor. Your loved ones, and eerie tom cruise videos, reanimate unease with deepfakes, Mar 2021.

[63] Alexander Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. Phoneme recognition using time-delay neural networks. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 37(3):328–339, 1989.

[64] Run Wang, Felix Juefei-Xu, Meng Luo, Yang Liu, and Lina Wang. Faketagger: Robust safeguards against deepfake dissemination via provenance tracking. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3546–3555, 2021.

[65] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8695–8704, 2020.

[66] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7):560–576, 2003.

[67] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8261–8265. IEEE, 2019.

[68] Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7556–7566, 2019.

[69] Xuejing Yuan, Yuxuan Chen, Yue Zhao, Yunhui Long, Xiaokang Liu, Kai Chen, Shengzhi Zhang, Heqing Huang, Xiaofeng Wang, and Carl A Gunter. Commandersong: A systematic approach for practical adversarial voice recognition. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 49–64, 2018.

[70] Tao Zhang. Deepfake generation and detection, a survey. *Multimedia Tools and Applications*, 81(5):6259–6276, 2022.

[71] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and simulating artifacts in gan fake images. In *2019 IEEE international workshop on information forensics and security (WIFS)*, pages 1–6. IEEE, 2019.

[72] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Two-stream neural networks for tampered face detection. In *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, pages 1831–1839. IEEE, 2017.