Breaking Security-Critical Voice Authentication

Andre Kassis* and Urs Hengartner†
Cheriton School of Computer Science
University of Waterloo
Waterloo, Canada
*akassis@uwaterloo.ca, †urs.hengartner@uwaterloo.ca

2023 IEEE Symposium on Security and Privacy (SP) | 978-1-6654-9336-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/SP46215.2023.10179374

***Abstract*—Voice authentication (VA) has recently become an integral part in numerous security-critical operations, such as bank transactions and call center conversations. The vulnerability of automatic speaker verification systems (ASVs) to spoofing attacks instigated the development of countermeasures (CMs), whose task is to differentiate between bonafide and spoofed speech. Together, ASVs and CMs form today's VA systems and are being advertised as an impregnable access control mechanism. We develop the first practical attack on spoofing countermeasures, and demonstrate how a malicious actor may efficiently craft audio samples against these defenses. Previous adversarial attacks against VA have been mainly designed for the whitebox scenario, which assumes knowledge of the system's internals, or requires large query and time budgets to launch target-specific attacks. When attacking a security-critical system, these assumptions do not hold. Our attack, on the other hand, targets common points of failure that all spoofing countermeasures share, making it real-time, model-agnostic, and completely blackbox without the need to interact with the target to craft the attack samples. The key message from our work is that CMs mistakenly learn to distinguish between spoofed and bonafide audio based on cues that are easily identifiable and forgeable. The effects of our attack are subtle enough to guarantee that these adversarial samples can still bypass the ASV as well and preserve their original textual contents. These properties combined make for a powerful attack that can bypass security-critical VA in its strictest form, yielding success rates of up to 99% with only 6 attempts. Finally, we perform the first targeted, over-telephony-network attack on CMs, bypassing several known challenges and enabling a variety of potential threats, given the increased use of voice biometrics in call centers. Our results call into question the security of modern VA systems and urge users to rethink their trust in them, in light of the real threat of attackers bypassing these measures to gain access to their most valuable resources.**

I. Introduction

Automatic speaker verification systems (ASVs) are widely used for authentication where a claimed identity is verified by comparing features extracted from a given audio sample against a "voiceprint" obtained from previously collected recordings. ASVs have gained popularity, primarily due to their convenience and increased security compared to passwords, becoming the core component of voice authentication (VA)—a constantly growing multi-billion dollar market [1].

VA has been deployed in security-critical environments, such as banks (e.g., Citibank [2] and First Direct [3]). Nuance [4] (recently acquired by Microsoft for $19.7B), Aculab [5], and similar products are widely used for authentication in call centers or for enabling transactions using smartphone apps [6], [7]. In light of the mass adoption of VA and boastful statements ("No one else has a voice just like you" [8]) of its vendors, it becomes imperative to evaluate its security under realistic threat scenarios.

Several attacks against ASVs have emerged [9]. Yet, the popularity of VA as a robust authentication platform is still on the rise. The reason is that no existing attack has demonstrated a proven ability to circumvent VA under strict security-critical conditions. Spoofing attacks (or deepfakes), such as speech synthesis (SS) [10] or voice conversion (VC) [11], [12], have shown great potential in fooling ASVs via fake audio generated in the victim's voice [13]. However, this threat has been known for years [14], leading to the development and of spoofing countermeasures (CMs)—complementary systems deployed side-by-side with ASVs, whose task is to detect spoofed speech. CMs are now being deployed by VA industry leaders [15], such as Nuance [16] and ID R&D Live [17]. The high accuracy achieved by these systems [18] and their increased ability to generalize [19] disqualifies spoofing attacks. On the other hand, non-spoofing, adversarial attacks against ASVs (and automatic speech recognition systems—ASRs) fail in security-critical environments as they (unrealistically) assume whitebox access [20]–[22], are query-inefficient (hundreds or thousands to the target model) or time-inefficient [23]–[28], or are untargeted (cannot impersonate a *specific* user) [29]. Finally, while there have been several attempts to adversarially attack VA platforms by generating spoofed speech first and then adversarially modifying it to fool CMs [30]–[34], which is similar to the approach we take, these share the same limitations with the adversarial attacks above.

This paper presents the first practical attack on VA in **security-critical** environments. This is the only attack so far that meets all limitations of such environments—fully blackbox, model agnostic, real-time, query-efficient and, most importantly, targeted (the attacker can impersonate any chosen victim). Furthermore, our attack accounts for all entities present in the strictest form of real-world security-critical VA, which typically employs text-prompted (independent) schemes. That is, instead of fixed, replayable phrases, the user is prompted with a random phrase to repeat upon each access attempt. The entities to be circumvented by the attacker *simultaneously* are the ASV, the CM, and a speech-to-text (STT) unit verifying the random phrase was repeated correctly. We formalize security-critical environments and their components in §II-D. Our attack exploits the known vulnerability of ASVs to spoofing attacks [13] and *mainly targets the CMs*. Unlike the few existing ineffective attacks on CMs, our

DOI 10.1109/SP46215.2023.00139

attack employs signal-processing (optimization-free), model-agnostic transformations to grant spoofed speech the ability to universally bypass CMs, while preserving the readily-existing voiceprint and textual content to still bypass ASVs and STTs. As a consequence, users will lose trust in VA, and vendors will have to invest resources in improving its security.

Overall, this work makes the following contributions:

- **Our attack can circumvent any state-of-the-art VA platform implementing the authentication protocol in its strictest form and is fully black-box, query-efficient, real-time, model-agnostic (transferable), and targeted.** To commit fraud or identity theft, attackers in security-critical environments must mount *targeted* attacks and will not have access to the target model's internals nor will they be able to query it repeatedly. A key contribution of this work is the ability to generate spoofed audio that simultaneously overcomes all components found in security-critical VA in a model-agnostic manner, enabling, for the first time, the execution of targeted attacks. Unlike traditional adversarial attacks on VA, we target the CMs. Namely, our contribution is the discovery of multiple signal processing transformations that can universally make any CM accept spoofed speech, without degrading the quality of the sample. The optimization-free nature of our attack circumvents the lack of transferability and the query and timing limitations hindering traditional adversarial attacks on VA. Due to the known vulnerability of ASVs to spoofed speech, attacking the newly-identified weakest link (CMs) drastically compromises the security of VA.
- **Our attack is effective against the full VA stack.** Although our attack forces CMs to accept spoofed speech, the ultimate goal is to bypass the entire VA stack. Hence, transformed samples must bypass the ASV, which follows from the vulnerability of ASVs to spoofed speech, given that our transformations are subtle enough to preserve this property. Similarly, the contents of the transformed samples must match the text expected by the STT. Therefore, we design our attacks such that the changes to the spoofed speech are minimal. We verify these assumptions through large-scale experiments involving 14 state-of-the-art CMs, five renowned ASVs (including a commercial system—Amazon Connect Voice ID [35]), and two commercial STTs [36], [37], proving them all vulnerable. With only up to six authentication attempts, the attacker achieves a success rate against the full authentication stack as high as 99%. Our user study further proves that even a human inspecting the outputs of our algorithm will be fooled. Finally, we establish that our attack remains effective even when executed over the phone network. To the best of our knowledge, this is the first-ever *targeted* adversarial attack over the phone.
- **Robustness to various defenses:** We analyze the potential of different defenses in mitigating our attack. While we find adversarial defenses limited, some other adaptive techniques seem more promising. We evaluate these techniques and find that, when combined, they can degrade the attack's performance. However, the attack remains effective even after this degradation occurs.
- We make our source code publicly available to aid other researchers in the construction of future defenses[1].

II. Background

A. Automatic Speaker Verification (ASV)

At the core of ASVs lies the voiceprint—a set of unique features of an individual found in their voice. This voiceprint captures physical factors like the shape and size of the vocal tract and larynx, and a "behavioral signature" consisting of one's accent, rhythm, pronunciation and more [38]. This led to investigating voiceprints as biometrics.

ASVs operate in two phases: enrollment and verification. During enrollment, the user supplies speech samples used to extract their voiceprint, which serves as their identifying signature onward. Upon future verification attempts, the user's identity is verified via a provided speech sample x together with the claimed user identity U_{ID} to whom the sample supposedly belongs. The sample is checked against the voiceprint outputting a decision (accept/reject). The notation $DB(UID)$ refers to a database query on the server side with the provided identity that retrieves the voiceprint, and $ASV(x,\ DB(U_{ID}))$ indicates that the ASV accepts x with respect to the voiceprint of U_{ID}. Verification can follow one of three schemes: 1) Text-dependent—a fixed phrase is repeatedly used, 2) Text-independent—the user may provide any random phrase, or 3) Text-prompted—the system requires a specific random text from the user to be spoken. The second and third options are more secure due to their robustness to replay attacks.

Researchers have discovered various sets of features for ASVs, including the Mel-frequency Cepstral Coefficients (MFCC) and Log Power Magnitude Spectrum (LPMS). ASV architectures vary significantly, but mainly fall under three categories: GMM i-vector systems, x-vectors, and DNN end-to-end approaches (operate on raw waveforms) [39].

B. Spoofing Countermeasures (CMs)

ASVs are vulnerable to spoofing attacks [13], which are classified into four categories: 1) Mimicking [40]—manipulating one's voice (without machines) to sound like the victim, 2) Speech synthesis (SS) [10]—using technology to produce samples in the victim's voice, 3) Voice conversion (VC) [41]—using technology to convert one's voice into the victim's, and 4) Replay [42]—replaying a previously-recorded sample of the victim. ASVs are robust to mimicking attacks [43]. Therefore these are not a real threat. Replay can be avoided using text-independent/text-prompted schemes. However, VC/SS attacks pose serious threats to VA [13].

The need for reliable authentication led to the emergence of spoofing countermeasures (CMs) [14], [44]. These are classifiers solving the problem of distinguishing between human

[1]https://github.com/andrekassis/Breaking-Security-Critical-Voice-Authentication

(bonafide) and machine (spoofed) speech. The literature distinguishes between active and passive CMs [45]. Active methods (e.g., VoiceLive [46], VoiceGesture [47], EchoVib [48]) rely on heuristics and sensory data to capture liveness cues that accompany human speech, such as articulatory gestures or the shape of the vocal tract. Passive methods focus on measurable differences in the audio waveform itself that differentiate human from machine speech [19], [44], [45]. Active methods are much less practical since they are sensitive to sensor and microphone placement and device-dependant, while passive methods can be integrated into existing VA software [45], which is done by pioneers in the filed such as Nuance [16] and ID R&D Live [17]. Our focus is on passive CMs.

Not every VA system is security-critical. For instance, Siri and Google Assistant neither enforce strong security measures nor use CMs. However, leading VA providers, such as Nuance, Aculab, and ID R&D Live, and others are explicit about deploying CMs to defend against spoofing as integral components of their systems [5], [16], [17]. Thus, successful attacks against security-critical VA must circumvent CMs.

C. Speech-to-text Systems (STTs)

Known also as automatic speech recognition (ASRs), STTs are used to transcribe spoken phrases into text. In the context of VA, STTs are coupled with ASVs to implement text-independent/text-prompted schemes and defend against replay attacks. The STT is used to transcribe the user's input (sample) and the transcription output is examined against the expected text. The STT accepts the authentication attempt only if the contents match. Given a spoken phrase x and text t, the notation $STT(x,\ t)$ indicates that STT accepts x w.r.t t.

D. VA in Security-Critical Environments

Security-Critical Environments (e.g., banks) manage access to highly sensitive resources and employ *strong* authentication protocols to enforce the following:

Limited number of failed attempts. Multiple failed attempts must lead the target to stop accepting further attempts. Attacks on (voice) authentication platforms that ignore this requirement are invalid. For instance, credit cards are invalidated if the user enters their PIN incorrectly 3 times, and Apple locks iPhones after 6 incorrect pass-codes.

Limited response time. The user (attacker) has to generate authentication responses and transmit them within seconds.

No access to system internals. The system is blackbox returning only decisions (accept/reject), as assumed in research [25] and practice [16], [17]. We do not assume more permissive settings where systems reveal output probabilities.

Success under limited time and attempt constraints sets us apart from all previous blackbox attacks on VA (see §X), while attacks that cannot operate in blackbox settings are not practical. Our attack is fully blackbox, takes 4s on average, and is highly successful with only three attempts.

Text-prompted/independent scheme. The recommended form of VA uses text-prompted/independent schemes to mitigate replay attacks. Some vendors may use text-dependent schemes with fixed phrases, which are vulnerable to replay attacks. However, there are CMs that detect replayed speech [44]. Hence, we assume that simply replaying a recording is not a valid attack and focus on more restrictive text-prompted/independent schemes.

All in all, for successfully passing an authentication attempt in a security-critical environment with a phrase x given identity U_{ID} and text t, the following must hold:

$$ASV(x,\ DB(U_{ID})) \wedge CM(x) \wedge STT(x,\ t).$$

III. Threat Model

A. Definitions

Attacker. The person aiming to bypass a security-critical VA platform via adversarial or spoofed audio to pose as the victim.

Victim. The user protected by the VA system whose identity is at risk of being stolen by the attacker.

Target. The VA system (or any of its components—ASV, CM, or STT) the attacker wishes to fool.

B. Attack Channels

Here we list the common channels users (attackers) typically use to communicate with a remote security-critical VA system.

Designated-app. The user (attacker) issues a transaction using an app (e.g., a banking app) and authenticates it using their voice (instead of a password) [16]. The authentication phrase is passed to the server over TCP (i.e., reliably, without packet loss, jitter or lossy encodings [4]).

Over-telephony. The user (attacker) calls the service provider and interacts with the interactive voice response (IVR) system, which uses VA for identity verification [16]. The input is transmitted over the phone, introducing additional challenges, as attacks have to withstand the noisy medium (see VII).

C. Attacker's Goals, Knowledge, and Capabilities

Goals. The attacker's goal is to produce an audio sample capable of bypassing a VA system deployed in a security-critical environment. Based on the components we introduce in §II-D, the attacker is concerned with realizing an algorithm $\mathcal{A}$, whose inputs are text t, and the victim identity U_{ID} and whose output is an adversarial $x_{adv} = \mathcal{A}(t,\ U_{ID})$, s.t:

$$ASV(x_{adv},\ U_{ID}) \wedge CM(x_{adv}) \wedge STT(x_{adv},\ t).$$

The system accepts verification phrases in one of the two methods described in III-B. The attack must comply with the protocol described in §II-D.

Optional goal: We optionally require that the samples provided be robust to human inspection. In security-critical platforms, authentication attempts can be recorded for future audit purposes [4] (e.g., disputing suspicious transactions) during which the recordings may be inspected by humans. Passing the human judge's (HJ) inspection, whose task is to listen to the sample **in case of an investigation** and verify whether it in fact comes from the victim, is indicated as $HJ(x_{adv},\ U_{ID})$.

This goal is optional as the attacker may ignore it if an investigation is not a concern (i.e., if the attacker covers their traces or can guarantee that the transaction will go through before any flags are raised—for instance, when stealing typical

amounts for the victim to transfer). Our attack satisfies this goal, as opposed to traditional adversarial attacks against ASVs, which do not produce samples in the victim's voice.

Knowledge. We are concerned with the most realistic black-box setting, where the attacker only has access to the target's decision (accept/reject), without knowledge of the target's internals, training data or output probabilities.

Capabilities. *Access to a rooted phone (designated-app only):* The attacker can bypass the microphone on the device used to interact with the target to inject audio directly into the app. This requires that the attacker obtains *some* rooted device. We do *not* assume access to the victim's device. Rooting is possible as rootkits are available online [49]. Apps may be equipped with "root detectors", but these can often be evaded [50]. The ability to bypass the microphone has been assumed in many previous works [21]. This is not required for over-telephony attacks, as the attacker can use a computer. *Access to audio recordings of the victim (designated-app & over-telephony):* The attacker is in possession of sufficient data (and computational resources) to train SS/VC models in the victim's voice. Modern SS/VC systems accomplish this with only ∼15 minutes of victim speech [51]. Such data is easy to obtain thanks to social media and conference calls, especially for a determined attacker targeting a specific individual. Alternatively, zero-shot algorithms (which we do not consider) only require 5 seconds in the victim's voice (but only retain a 50% success rate against human listeners [13]). *Obtaining PINs (designated-app only):* The attacker (pretending to be the victim) installs the app the victim uses to interact with the target on the attacker's rooted device. Most banks allow their users to install their apps on multiple devices. They may require entering a PIN [52], which the attacker can obtain through social engineering [53]— a standard assumption [26]. In the over-telephony scenario, this requirement is waived, as the attacker may use any phone (even a landline) to place the call, and no app (therefore no PIN) is required.

IV. Attacking a Security-Critical VA System

A. Attack Overview

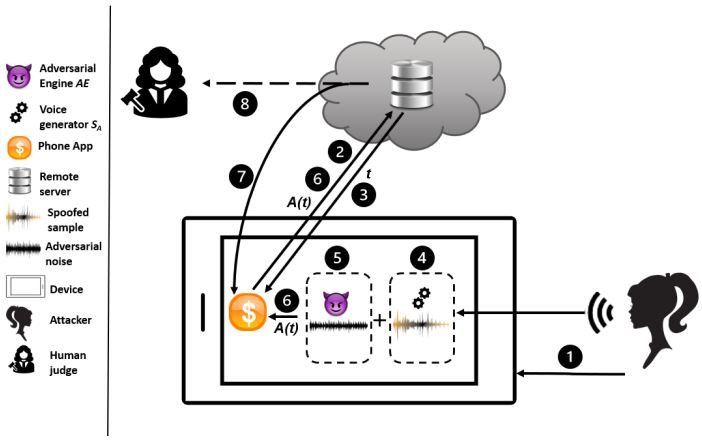

Figure 1: Attack Overview.

Our attack is motivated by two main hypotheses:

H1: *There exist **identifiable**, **removable**, and **universal** key differences between human and machine speech.*

H2: *These nuances are among the **primary** telltales on which CMs rely to make their decision.*

Note: The term universal above implies that these differences (nuances) are agnostic to the user or spoken text.

Optimization-free universal attacks on ASVs are hard to realize as they require fathoming features that characterize the victim's voice and how they can be injected into different utterances. This is evident by the variety of target-specific, non-universal attacks against those systems (see §X), and absence of universal attacks, despite the recurrent emphasis on their importance [25]. However, based on ***H1***, which states that there are human-understandable universal differences (across users and speech samples) between machine and human speech, and ***H2***, which states that they are crucial for CMs to perform correct classification, we identify a new potential weakest link. Successfully eliminating those intuitive differences in a universal manner for a given spoofed sample through designated transformations can make the components protecting ASVs against spoofing (i.e., CMs) misclassify. This will leave the entire VA platform vulnerable, due to the known vulnerability of ASVs to spoofing attacks. Furthermore, the nature of the targeted nuances entails that defending against our attack will not be trivial, as we target core CMs features, leaving them practically useless. The novelty in our work lies in the realization of these transformations that target commonalities of various CMs without resorting to transferability, which is of limited success (see §VI-A). The challenge is that these transformations need to be subtle to preserve the properties of the spoofed sample that make it effective against ASVs, and its textual content (necessary to bypass STTs).

Overall, the attack is as follows (see Figure 1): 1) The attacker accesses the target-provided app (or places a call to the target) and issues a command triggering a transaction. 2) The target receives the request and 3) prompts the attacker with a random authentication phrase t. 4) The attacker generates the requested phrase in the victim's voice via an SS or VC algorithm S_A. 5) The result is forwarded to the adversarial examples engine AE, which perturbs it using our transformations to generate a sample with identical content in terms of its text, while also preserving the voiceprint of the victim. 6) The adversarial example is injected into the app (call) and transferred to the target. 7) The target verifies that the sample bears the victim's voiceprint (ASV accepts due to the vulnerability of ASVs to spoofing attacks), the textual content of the sample is t (STT accepts), and the sample comes from a human (CM accepts). 8) If the transaction is later reported as suspicious, the human judge is fooled by the seemingly genuine sample and confirms the transaction's authenticity.

B. Universal Transformations Against CMs

We seek transformations than can eliminate identifiable and removable machine cues, while preserving quality. Hence,

these transformations must be crafted based on an understanding of the design of CMs and fake speech algorithms. We surveyed numerous papers to amass such an in-depth understanding, leading to the development and testing of various heuristics. Below we list the selected transformations:

F_1. *Replacement of Leading and Trailing Silence:* When humans speak into a microphone, there is static noise due to recording device imperfections (characteristics of the microphone), in addition to involuntary sounds the speaker produces that accompany the speech (e.g., the speaker inhales before talking). Spoofed speech lacks these "non-speech" cues, which may influence the decision of CMs. In fact, applying voice activity detection (VAD) to filter out silences when training or evaluating CMs results in a performance degradation [54], proving these non-speech cues essential for CMs.

One may incorrectly assume that CMs can be trivially made ineffective by removing silence intervals, eliminating them as a deciding factor, and making CMs easier to fool [54]. However, when replicating these experiments, we find that eliminating silences makes CMs more likely to reject any input. Hence, this attack will not be successful (the attacker needs the model to accept spoofed samples) as the performance degradation when silence is stripped is the result of the inability of CMs to accept *bonafide* samples that lack the trimmed silence periods, which is of no use as spoofed samples are still rejected.

Nonetheless, "natural" silence can be copied from bonafide samples. F_1 replaces leading/trailing silence in spoofed samples with silence from benign samples. Appending longer silence intervals to increase the dependence on these cues is not a valid attack, as research shows trimming long silences possible, making them shorter but sufficient for classification [55].

F_2. *Elimination of Inter-word Redundant Silence:* F_2 further eliminates non-speech cues that identify spoofed speech by removing long "synthetic" silence intervals. SS/VC algorithms fail to generate statistically-natural silence to fill gaps between consecutive words. Spoofed silence intervals are often too quiet compared to human speech, since humans have air flow between their words. Similarly, natural speech exhibits an echo, as the different reflections of the same sound arrive at the microphone at different times (see F_4), populating those silent regions. Furthermore, spoofing algorithms often generate speech by dividing it into time steps, generating each independently. This step-by-step (often non-causal) procedure may cause the output to have extended unnatural silences, although the words at each step may have a (semi-)natural silence distribution between them. As opposed to leading/trailing silences, these cannot be replaced as their location mandates that the distribution of their static noise must be compliant with preceding and following intervals and replacing them with silence from different samples will not accomplish this. Thus, F_2 eliminates inter-word silences altogether. We find that since F_1 already injects human-like silences, in F_2 it is sufficient to remove redundant silent periods. Using the right parameters, we reduce only long (empty) silence intervals but do not eliminate them completely, so the output still contains the more lively (noisy) silences if they exist, sounding natural in terms of the speed at which words are uttered.

F_3. *Boosting the center of the spectrum:* The majority human speech energy is concentrated in lower parts of the spectrum. Thus, our hypothesis is that this frequency range naturally becomes the most critical in audio-related tasks compared to negligible higher frequency components. Furthermore, many audio classifiers (CMs, ASVs, ASRs) rely on tempo-spectral representations that emphasize details in lower frequencies while de-emphasizing high components (e.g., MFCC [10]) or have higher resolution at lower frequencies (e.g., CQCC [56]). CMs often use such front-ends. Moreover, the latest state-of-the-art CMs operate on raw waveforms. Due to the energy distribution of speech (concentration around lower frequencies), we expect these systems to naturally assign larger weights to lower components as well. Thus, CMs are more sensitive to cues in lower frequencies. Introducing perturbations manipulating the spectra s.t higher frequencies are amplified relative to these lower components decreases the magnitude of the cues that can be more precisely classified by CMs, while the regions that are less accurately learned will be made dominant. The lack of accuracy in this range causes misclassification.

This transformation amplifies the intensity of the signal's spectrum around the target region. The range $(1-4)KHz$ has the most considerable effect on speech intelligibility [57]. Higher frequencies (typically above 6KHz) are dominated by noise, less important to intelligibility, and are naturally of lower magnitudes. Thus, we ensure keeping these higher frequency components suppressed by using a sharp (fast-decaying) filter around the center of the spectrum.

F_4. *Local Echo:* This is an extension to F_2. As opposed to spoofed speech, natural speech is typically recorded with a microphone, which receives its input from a user speaking into it. The user's voice does not travel to the microphone in a single trajectory. Instead, voice from their mouths travels in many directions and gets reflected at different angles and times, echoed back to the microphone. With time, the amplitudes of the reflected particles belonging to each utterance dissipate, no longer being recorded. The lack of echo in spoofed speech can serve as a sign telling benign and spoofed samples apart. F_4 generates local echo, introducing 1ms-shifted copies to the signal over a short period (100ms), decreasing the amplitude of the repeated copy as we advance in time.

F_5. *Pre-emphasis:* This transformation applies pre-emphasis to the input signal. Pre-emphasis [58] is used to alter the signal at different frequencies by amplifying the relative magnitudes of higher frequency components at the expense of lower frequencies. It is popular in high-speed transmission systems, creating noise-robust signals before transmission. Upon receipt of the signal, the operation is inverted via de-emphasis.

The intuition behind using pre-emphasis is similar to that motivating F_3— the uncertainty of CMs at higher frequencies. Yet, a drawback of pre-emphasis is that its frequency response is monotonically increasing, leading to large magnitudes at very high frequencies, and yielding outputs with flat spectra (the energy is evenly distributed along the frequency axis).

Those are unnatural compared to normal speech, wherein the energy is concentrated at lower frequencies, and will sound unrealistic. Instead, as in F_3, the target region we wish to amplify is the center of the spectrum (around 1KHz-6KHz).

To circumvent this, we use a smaller coefficient for the operation (0.5) instead of typical large coefficients (0.97) to ensure the output signal complies with the spectral map of natural speech. This limits the usability of F_5, although it is still highly effective. F_3 can, hence, further boost the attack's performance as it enables defining the specific region to amplify, leaving out undesired frequencies. Yet, F_5 enables us to systematically manipulate the entire spectrum as a well-established technique, preserving inter-connections between different frequency components, which is harder with F_3.

F_6. *Noise Reduction:* The samples produced by spoofing algorithms often exhibit machine noise, which characterizes the algorithms used to produce them. In a way, this noise can be thought of as a fingerprint attesting to the algorithm's imperfections. As opposed to naturally-occurring (additive) noise, it does not exhibit the same normal distribution over the spectrum, and will be centered around specific frequencies, forming a cue that CMs may spot. This computational limitation is most likely due to dataset biases, as the settings under which the training data is gathered (e.g. recording devices) may introduce such artifacts. If such artifacts are eliminated, CMs will lose the ability to rely on them to output a decision.

We implement F_6 via spectral gating [59], which operates by first calculating the tempo-spectral representation of the signal. Afterward, for each frequency, the mean and standard deviation over time are calculated. Lastly, at each time step, if the intensity at some frequency is not sufficiently larger than the mean for that frequency over time, this component is eliminated. This filters out continuous background noise at each frequency, leaving only unique, audible (non-noise) information. Additionally, we find introducing additive noise at this layer replaces the mechanic with natural noise found in human samples and enhances the attack.

F_7. *Adversarial speaker regularization (ADVSR):* We include a final optimization-based layer ($ADVSR$). However, this does not change the optimization-free, model-agnostic nature of our attack, since we do not use a shadow CM to perform the optimization. The optimization is done via a model designed for an adjacent task—an ASV. The objective of this step is not to achieve high success against target ASVs, nor do we rely on transferability. The goal is to utilize a shadow ASV to optimize a spoofed sample to increase its ability to fool CMs. Hence, this step is still considered a model-agnostic transformation. ASV-based optimization can eliminate machine artefacts. The idea is to "engrave" the user's voiceprint into the spoofed sample. The rationale is that given a reference sample y by the victim user, and the spoofed sample x, a non-zero ASV loss (for x w.r.t y), indicates that there is some uncertainty as to whether the speaker in both samples is the same, and we attribute that to one of the samples being "spoken" by the machine. When the ASV removes these differences, machine cues are removed as a byproduct making the sample less machine-like and capable of bypassing the CM.

Aside from $F_1 - F_7$, we experimented with various other techniques (excluded for limited success), including CM-based adversarial attacks (see §VI-A), vocal tract copying [60], and marginal filtering at fundamental frequencies and harmonics. We leave exploring additional transformations to future work.

C. Attack Implementation

First, we replace machine silence with "realistic" silence segments. We use $Librosa$ [61] to trim leading and trailing silence, with a threshold of -25.0 db, and replace it with silence from a bonafide random sample of the same speaker (F_1). F_2, uses $WebRTCVad$ [62] to remove inter-word extended silences (aggressiveness=3), with a frame of $30ms$. We find that this algorithm and hyperparameters retain necessary silence intervals and do not eliminate them completely, satisfying the requirements of F_2. We keep the original sample length after steps F_1 and F_2 by choosing bonafide sample periods that match in length those that F_1 and F_2 trim.

Afterward, we apply F_3 by computing the FFT of the input and amplifying the frequency range 1-6KHz by 1.5 using a sharp Butterworth BPF (with an order of 20). This limit of 1.5 ensures that the outputs do not sound suspicious to humans and avoids violating the spectral map of normal speech (see IV-B). F_4 is introduced by adding shifted copies (by $1ms$ at a time) with decreasing amplitudes. Assuming the original amplitude is A, the amplitude of the i'th shifted copy is $A/(8 * i + 1)$, since the longer it takes the echo to reach the microphone, the more energy it loses. Next, we apply F_5 based on $Librosa$'s implementation with a coefficient of 0.5. In F_6, noise reduction is applied [59] and additive noise is introduced instead with a limited amplitude of $\epsilon = 0.0035$ (to avoid generating noisy samples). To preserve the effects of F_1, we apply noise reduction between the leading and trailing silences only. Finally, we perform F_7 with the ASV described in §V. Since the ASV's loss requires a reference voiceprint of the victim, we use multiple bonafide (victim) recordings. We generate the adversarials using I-FGSM [63] and $\epsilon = 0.0015$.

The transformation order and parameters were determined empirically and can be explained as follows: F_1, F_2 are order-invariant, operating on different regions. F_3, F_5 are order-invariant; both perform (frequency) *amplification* operations. F_1, F_2 replace signal fragments and should precede F_3, F_5 to avoid discarding some of their effects. F_6 introduces noise that must be minimal (inaudible). Hence, it is applied after amplification (F_3, F_5). F_7 must be last since it is a volatile voiceprint-based transformation, while others are voiceprint-agnostic (destructive if applied afterward). F_4's (echo) placement (determined empirically) must be between F_2 and F_7.

V. Experimental Setup

This section presents the models used, their baseline performance, and our datasets.

System specifications. All adversarial examples were generated on a server running Ubuntu 20.04 on a PPC64LE, POWER8NVL 2.33 GHz CPU with 16 physical (128 virtual)

System	Front-end	EER (ASVspoof2019 LA eval.)
WAV2VEC [19]	Raw-audio	0.82*% (ASVspoof2021 LA eval.)
AASIST [66]	Raw-audio	0.83%
AASIST-L [66]	Raw-audio	0.99%
RAWGAT_ST (mul) [67]	Raw-audio	1.06%
SSNet [68]	Raw-audio	1.64%
RAWDARTS [69]	Raw-audio	1.77%
MCG [70]	CQT	1.78%
LCNN_LSTM [71]	LFCC	1.92%
MLCG [70]	CQT	2.15%
AIR [72]	LFCC	2.19%
Res2Net [73]	CQT	2.50%
AIR_AM [72]	LFCC	3.26%
DARTS [74]	LFCC	4.96%
LCNN [75]	LPMS	5.06%

Table I: Top-performing CMs with code and pre-trained models as of October 2021. In the first row, WAV2VEC is the state-of-the-art w.r.t the ASVspoof2021 LA evaluation set.

cores and 1TB RAM and 4 Tesla P100-SXM2 GPUs with 16G of memory each (although our experiments used one GPU at a time). To mount our over-telephony attack, we set up a dummy server as an Amazon Connect [64] instance and used Twilio [65] to programmatically place calls to it over which the samples were delivered. The samples were recorded at the receiver in AWS, downloaded to our server, and fed into the CMs.

Sampling & encoding. All samples in our datasets are sampled with a rate of 16KHz and represented as FP arrays in the range **[-1.0, 1.0]**— a standard representation (we normalize our transformations' outputs to this range). When transmitting samples over the phone, due to bandwidth limitations of the carrier (Twilio), they are received at a sampling rate of 8KHz, which we then up-sample to 16KHz to feed into our targets.

Systems. We select top-performing models from the literature or the industry, depending on availability.

CMs. Commercial CMs, such as Nuance's, are proprietary and closed-sourced, without public APIs. We reached out to leading platforms, such as Aculab and ID R&D Voice, but never heard back. Thus, to evaluate our attack against the state-of-the-art, we use open-source systems from the ASVspoof2019 challenge [76]. We focus on these since the challenge has long been the platform where novel CMs emerge. The challenge received contributions from renowned corporations (e.g., Google, which contributed to the challenge's dataset [77]), demonstrating its importance by industry standards. We do not restrict ourselves to systems that actively participated in the challenge and include more advanced CMs that have since appeared demonstrating superior performance (w.r.t the challenge's task). We consider the best reported models [66], [67] (released 2021), and for which the authors provide pre-trained models and results (that we verified)—see Table I. We use pre-trained models to avoid introducing any errors while reproducing them. These models' performance is on par with leading commercial systems. Specifically, the world-leading ID R&D Voice [17] [15] was the winner of the ASVspoof2019 challenge, achieving an equal error rate (EER) of 0.22%. The also proprietary system ranking second achieved an EER of 1.86% [76]. Our chosen systems achieve similar numbers, with some ranking between the top two contestants.

Since the list in Table I was compiled, new systems have emerged. Therefore we include a representative of such potentially more robust models. We use the state-of-the-art WAV2VEC [19], a model designed for the ASVspoof2021 challenge, whose objective was to build generalizable CMs robust to the audio channel effect (can correctly classify samples even over phone calls). The model was evaluated on the ASVspoof2021 evaluation dataset, which augments the ASVspoof2019 evaluation subset with samples transmitted over the phone. This model is capable of operating on clean audio (as found in the ASVspoof2019 dataset) and phone audio and is so far the best at performing both tasks without needing further calibration [19]. Note that WAV2VEC achieves the lowest EER (first row in Table I), and that this EER is w.r.t the more challenging, augmented ASVspoof2021 dataset. Since WAV2VEC is a very recent model, we only include it in our final experiments that evaluate our full attack.

Due to the known possible lack of generalization [14] to the over-telephony setting, some CMs may not be appropriate for this setting. This does not apply to WAV2VEC, which was specifically designed and evaluated for this task. Thus, in the over-telephony experiment, we naturally include WAV2VEC. For other CMs, we determined which can generalize to this setting and calibrated their decision thresholds accordingly. Details are in Appendix §B. In conclusion, we only found seven other models to be somewhat generalizable, although their performance is inferior to WAV2VEC.

Our chosen CMs vary significantly in their front-ends (see Table I) and architectures, which proves our attack bypasses any VA platform.

ASVs. We use various architectures to represent the state-of-the-art (GMM i-vectors, x-vectors, and end-to-end). Using KALDI [78], we design three GMM i-vector models of different dimensions and front-ends (ASVShadow, ASVTarget1, ASVTarget2), and a single x-vector model (ASVTarget3). KALDI is widely used in academic and industrial projects and was used to evaluate previous works [79]. Additionally, we use Resemblyzer [80], which implements Google's GE2E [81] as an end-to-end ASV. Resemblyzer is popular and has been used in academic works [13]. Finally, we include the commercial Amazon Connect Voice ID [35], which is available for call centers using Amazon Connect [64] to verify callers' identities. The ID delivered in each call was used to invoke Voice ID with the (previously-uploaded) voiceprint of the speaker to whom the delivered sample is claimed to belong. Each voiceprint consists of 30 seconds of speech, obtained by concatenating bonafide samples from the ASVspoof2019 evaluation set for each speaker. At the time these experiments were conducted, Voice ID was not using a CM. Details of all ASVs are in Appendix §A. We used ASVShadow for F_7. All other models were used as targets. The difference in architectures and training data preserves blackbox assumptions.

STTs: We use two commercial, widely-used (blackbox) STTs: Google Cloud STT [36] and Microsoft Azure STT [37].

Training datasets. *CMs:* We mainly experiment with pre-trained CMs developed for the ASVspoof2019 challenge,

which (except WAV2VEC) were trained on the training subset of the LA portion of the ASVspoof2019 dataset [82]. It includes samples from 107 speakers (46 males/61 females). For each speaker, there are bonafide and spoofed (VC/SS) samples. The dataset is partitioned into 3 sections: training, development, and evaluation. The training and development datasets contain samples from 20 speakers each, while the evaluation samples come from 67 speakers. The subsets are disjoint in speakers and spoofing algorithms. For WAV2VEC, the pretrained model uses the ASVspoof2019 training subset, augmented with additive and convolutional noise.

ASVs: Training the GMM i-vector ASVs was done using the bonafide samples from the (disjoint) development and training subsets of ASVspoof2019 for the shadow and targets, respectively. The x-vector system was trained using Voxceleb1 [83], augmented using MUSAN [84]. Resemblyzer and AWS Voice ID come pre-trained. To use them, their thresholds must be calibrated. We generated calibration data from the evaluation subset of ASVspoof2019 (see Appendix §A for details).

STTs: We used pretrained API-accessible systems.

Evaluation dataset. Spoofed samples in the ASVspoof2019 evaluation subset were generated by various state-of-the-art spoofing algorithms (as of 2019). As the output quality of SS/VC models depends on the availability of training samples and capabilities of the algorithms, we find samples in this subset to be of different qualities. Since our attack optimizes high-quality spoofed speech (such as [10]), we decided to focus on a reduced set of this subset, including only samples of the expected quality that can initially bypass STTs and fool humans. This decision is further motivated by the need to have a reduced set to evaluate in our large-scale experiments (especially in the costly and time-consuming over-telephony setting). After listening to a few thousands of recordings, we chose 474 spoofed samples belonging to 48 users, generated by 13 different algorithms we deemed of satisfactory quality. Compared to previous works considering over-telephony adversarials [29] and spoofed speech [85], this number is similar and the variety in users and algorithms is more extensive. To promote future research, we make our high-quality reduced subset available[2]. We also include the spoofed and adversarial samples we use in the user study in §VIII.

Evaluation metrics. We define our attack's success as the Acceptance Rate (AR) for each system type. For CMs, this is the probability that the sample is deemed bonafide. For ASVs, this is the probability that given a reference victim voiceprint, the ASV's comparison of the attack sample against this reference leads to acceptance. For STTs, acceptance indicates a successful comparison of the transcribed text from the sample against the phrase expected by the system. We assume spoofed samples always satisfy this property, due to the quality of modern audio spoofing. For adversarial examples, acceptance is the transcribed texts from the original sample and the adversarial generated from it matching.

[2]Available at the GitHub repository accompanying this work

VI. Attacking Designated-App Authentication

This section focuses on the designated-app scenario (see §III-B), where authentication phrases are passed to the server as waveforms over TCP (i.e., reliably, without packet loss, jitter or lossy encoders [4]). First, we show results against each of the CMs, ASVs, and STTs independently. Then, we present the joint results for them combined. Recall that we target CMs—success against STTs/ASVs is due to the quality of the used spoofed speech, the vulnerability of ASVs to it, and our attacks' ability to preserve these properties.

A. Results Against CMs

Shattering the myth of transferability: The reader may question the need for our transformations given the existence of conventional adversarial techniques. Assuming a query-limited attacker leaves us with transferability attacks (crafted against a shadow CM and transferred to the target), as done in previous works [30]–[33]. However, transferability was proven ineffective for other audio-related tasks [86]. We prove that it is not a viable attack against CMs either, pinpointing flaws in previous works. Details are in Appendix §C.

Our transformations: The collective (cumulative) attack results are in Table II. The transformations should be performed collectively as they are motivated by various hypotheses and target different cues. We report individual evaluations of each transformation in Appendix §D. In a nutshell, each transformation individually incurs a significant increase in the attack's success against several (but not necessarily all) CMs compared to spoofing baselines. Nonetheless, these do not affect all CMs similarly as they employ various heuristics, leading to a minor degradation w.r.t the baselines on rare occasions. The ensemble attack (discussed below) remains universal, far outperforming singletons.

For F_1 (silence replacement), compared to spoofed samples alone without any additional transformations (first row in Table II), it is clear that F_1 leads to a considerable degradation in the performance of several CMs. Even CMs that were completely unaffected by additive noise or transferability (RAWGAT_ST, AASIST, AASIST-L—see Tables VIII & IX in Appendix §C) exhibit a performance degradation under this simple transformation. This is evidence that silence intervals of bonafide recordings encode non-speech liveness cues missing in machine speech. These non-speech cues can be injected by replacing silent intervals with genuine silences, as done by F_1.

Next, we study F_2 (silence elimination). As explained in §IV-B, additional non-speech cues guiding CMs are embedded in inter-word silence intervals, as SS/VC algorithms often fail to mimic genuine silences, generating long, mute segments. This is cemented by F_2's findings, demonstrating a significant increase in the attack's success against all CMs.

F_3 (spectrum center boosting) is also highly effective. We can see that increasing the relative amplitudes of the uncertainty regions at the expense of lower frequencies (see §IV-B) drastically increases the success rates.

The results for F_4 (local echo) indicate that the contribution of this transformation is only incremental compared to the

Frontend	LPMS	LFCC				CQT			Raw Waveform				
System / Transformation	LCNN	AIR	AIR_AM	DARTS	LCNN_LSTM	MCG	MLCG	RES2NET	SSNET	RAWGAT_ST	RAWDARTS	AASIST	AASIST-L
None	3.16%	2.32%	3.59%	4.01%	2.95%	1.90%	4.01%	2.95%	2.74%	3.16%	1.90%	2.32%	2.32%
F1	13.71%	15.61%	13.92%	13.08%	10.97%	12.66%	9.28%	4.85%	12.66%	7.59%	1.69%	4.85%	5.27%
F1-F2	19.41%	23.00%	17.30%	16.03%	15.40%	16.24%	15.82%	4.22%	41.56%	9.70%	3.38%	13.29%	10.97%
F1-F3	20.25%	38.40%	28.90%	14.98%	24.89%	32.70%	20.04%	7.59%	43.46%	9.92%	4.64%	15.40%	13.08%
F1-F4	19.83%	39.87%	30.17%	16.46%	24.89%	34.60%	20.46%	8.44%	47.05%	10.76%	5.49%	15.82%	14.35%
F1-F5	**21.31%**	44.30%	34.39%	20.46%	24.05%	45.99%	27.85%	8.02%	54.01%	11.81%	12.03%	18.99%	17.93%
F1-F6	12.03%	45.99%	42.19%	**20.68%**	**25.11%**	**67.51%**	45.15%	**16.03%**	57.81%	11.18%	15.82%	24.68%	21.73%
F1-F7 (Full Attack)	17.93%	**58.02%**	**51.27%**	20.04%	15.61%	65.82%	**52.74%**	15.82%	**61.81%**	**12.03%**	15.82%	**24.89%**	**21.94%**

Table II: Attack results against CMs. Bold entries represent the best success results for each system.

other transformations. Nonetheless, we still see some improvement over the cumulative attack without F_4 for the majority of CMs, especially for the harder systems to spoof, such as RAWGAT_ST and RAWDARTS. These systems have proven so far to be far more robust than counterparts and therefore any improvement is significant and proves the reliance on this echo as a cue. For instance, this is the first time we see a success rate of above 5% against RAWDARTS.

F_5 (pre-emphasis) is among the leading transformations, especially against CMs operating on raw waveforms. Recall that for the pre-emphasis filter, to ensure the output sample's spectrum conforms to the spectral map of natural speech (see §IV-B), we chose a smaller constant than the typical. Interestingly, while large coefficients produce unnatural-looking (sounding) spectra failing human inspection, most CMs (except $LCNN$, $LCNN\_LSTM$, $DARTS$) are vulnerable to pre-emphasis with a large coefficient (0.97) (see Appendix §D). Not only will such samples likely fail human inspection, pre-emphasis with large coefficients can also be defeated by adding a component measuring the statistical distribution of the signal's spectrum to reject unnatural-looking samples. Comparing the spectral density against the natural distribution of speech is a vital step currently missing in many CMs. Nonetheless, our attack produces natural-looking examples, robust to this proposed defense.

F_6 (noise reduction) eliminates characteristic noise due to imperfections of spoofing algorithms that may be identified by CMs, replacing it with additive noise following the observation that bonafide samples often exhibit such noise. We see a significant increase in the success against most CMs (except $RAWGAT\_ST$), making F_6 among the most important transforms, and highlighting CMs' dependence on *removable* cues.

The last layer is ADVSR (F_7). When coupled with the rest of the transformations, it achieves the highest attack success rate for multiple CMs, cementing the reasoning behind it.

State-of-the-art (WAV2VEC): WAV2VEC was designed for channel generalization and trained on noise-augmented data, which was proven to increase noise robustness and improve generalization [19]. This is a natural defense against our attack, since the non-speech cues we target in F_1 and F_2 are linked to the absence of natural silence in spoofed samples. Additionally, our transformations boost central frequency components (F_3 and F_5). Although we boost these components that are of utmost import to intelligibility and CM classification, higher frequencies are amplified as a byproduct, which may also contribute to our attack's success. Higher components include noise-dominated information, which CMs should overlook (see §IV-B). WAV2VEC was designed to operate on telephony data, which often lacks components above 4KHz (see §VII). Hence, it should be able to disregard these components, while other CMs may be vulnerable to perturbations in this range.

Attack / System	ASVShadow	ASVTarget1	ASVTarget2	ASVTarget3	Google/Resemblyzer	AWS Voice ID
Spoofing	65.19%	66.24%	59.92%	99.79%	85.65%	69.15%
Our attack	100%	94.3%	81.86%	100%	75.1%	65.3%

Table III: Attack results against ASVs.

The spoofed samples have an initial success rate of 1.89% against WAV2VEC, making it the most robust CM we experiment with. After applying our transformations, the success rate increases to 11.6% (a 6.14x performance degradation), making it highly untrustworthy by the standards of a strict security platform. In conclusion, even the state-of-the-art is vulnerable to our transformations. The additive and convolutional noise used to train WAV2VEC does not necessarily capture microphone or articulatory sounds, which are the liveness cues found in the silence of bonafide speech, and therefore are less effective, leaving effective data augmentation an open challenge. Furthermore, WAV2VEC shares the vulnerability of being more accurate at lower (compared to central) frequency regions. Resistance in higher frequencies is found ineffective.

Summary: Our attack defeats state-of-the-art CMs, with a success rate of 11.6% against the most robust CM (WAV2VEC) and 65.82% against the most vulnerable (MCG). The main takeaway is that the cues these systems rely on can be easily spoofed. We also proved that adversarial transferability across CMs is an impractical attack strategy (see Appendix §C).

B. Results Against ASVs

Table III shows the success rates against ASVs. The first row shows the success of spoofed samples. Currently, there are even stronger spoofing algorithms that achieve higher success rates against ASVs. For instance, Wenger et al. [13] achieve a success rate of 100% against Resemblyzer, whereas our spoofed samples only attain a rate of 85.65%. Nonetheless, our spoofing attacks' rates demonstrate the vulnerability of ASVs to spoofing and the need for CMs. Our attack's success (second row) proves it retains the victim's voiceprint in the spoofed samples as it does not lead to considerable degradation in the success against any of the ASVs. The systems belonging to entirely different architectures (ASVTarget3, Resemblyzer) compared to our shadow (ASVShadow) in addition to AWS Voice ID, which the results imply is of a different architecture,

are almost insensitive to our transformations, maintaining high success. ASVTarget1 and ASVTarget2 exhibit a significant increase in the attack's success. We attribute that to our ASV-based adversarial layer (ADVSR), where ASVShadow is used to apply the optimization. Since all i-vector models share the same architecture (up to the variation in dimensionality), adversarial perturbations naturally transfer to the other two i-vector models (ASVTarget1/2) as opposed to different architectures. These results align with previous findings regarding the lack of transferability across different ASVs/ASRs [26].

We reported our findings about Amazon Connect Voice ID to Amazon, which has since deployed a CM. Evaluating our transformations against Voice ID's CM is left to future work.

C. Results Against STTs

To evaluate our attack w.r.t STTs, we use Google Cloud's [36] and Microsoft Azure's STT modules [37], and test whether 25 random adversarials generated using samples from our evaluation dataset are transcribed to the same content as the original samples. The success rates are 76% and 96% against Google and Azure respectively, (see Appendix §E).

D. Results Against the Combined Defense

We evaluate success against the combined defense components (CMs, ASVs and STTs). The human judge is only consulted after the fact in special cases (see §III). Thus, we exclude this component here. This is the most restrictive (and practical) setting under which VA attacks can be evaluated. We report success rates for 3 and 6 authentication attempts since these are typical numbers used in practice (see §III).

The results were calculated as follows: for each ASV-CM pair, we extracted the number of samples that bypassed the two systems jointly and divided it by the size of our evaluation dataset (474), yielding $\mathcal{P}_{\mathcal{AC}}$—the probability to bypass the ASV-CM pair. Given that our experiment with STTs only included 25 samples, we could not calculate the joint probability against all systems similarly. We assume the success rates against STTs to be independent of the success rates against the other components (since our attack is almost always successful against STTs)—in the future a more precise evaluation can be done. We use the average success rate against the two STTs (86%). Thus, the success rate against an ASV-CM-STT deployment becomes $\mathcal{P}_{\mathcal{ACS}} = \mathcal{P}_{\mathcal{AC}} * 0.86$. Finally, even though our transformations lack randomness (except ADVSR), we still assume that success at each attempt is independent of other attempts, due to the large space of spoofing algorithms and the infinite space of recordings of the victim that can be used to train them. Thus, success after n attempts has the probability $1 - (1 - \mathcal{P}_{\mathcal{ACS}})^n$.

The results are in Table IV. The combined systems indeed provide strong security against spoofing, with success rates as low as 0% even after 6 attempts (MCG/ASVTarget1). Yet, all combinations become unreliable against our attack, with the strongest (RawDarts/Voice ID) suffering success of 8.1% after 3 attempts and 15.55% after 6. For others, such as MCG/ASVTarget1, we approach 100% success. These results demonstrate how the strictest VA can be bypassed.

VII. Over-Telephony-Network Attacks

To the best of our knowledge, we are the first to explore the over-telephony setting for *targeted* adversarial attacks—the only known adversarial attack against VA over the phone is non-targeted [29]. We present superior attacks impersonating specific victims and fooling security-critical VA. Adversarial attacks over the phone are harder due to three main challenges: transcoding, packet loss and jitter [86]. An additional factor is bandwidth limitations. Samples are typically recorded with off-the-shelf devices at a high sampling rate. For instance, our inputs from ASVspoof2019 were recorded at 16KHz. Phone carriers typically operate at a lower rate (8KHz), causing information loss at high frequencies. Our goal here is to ensure that our attack can withstand these distortions.

We used the setup in §V to send samples over the phone. While Amazon Connect is robust to packet loss and jitter, the transcoding and bandwidth losses persist. As explained in §V, only WAV2VEC and seven other CMs are effective in the over-telephony setting and only they are considered. Table V shows the results of our over-telephony experiments involving the same 474 samples from §VI. The results clearly demonstrate the threats associated with our attack even in the over-telephony setting. Even the most powerful CM for the over-telephony task to date (WAV2VEC) is affected, bringing the success against it to 6.03% from 1.52% (3.97x). While the attack is less successful than in the app setting, it is still highly powerful especially after multiple attempts.

VIII. Fooling a Human Judge

In our threat model, attacks must fool humans (HJ) if they are asked to inspect the samples (see III). Given an audio sample and a claimed user identity (to whom the sample supposedly belongs), HJ verifies that the sample comes from that user. Since HJ is not necessarily acquainted with the user, this is done via a comparison to samples previously provided by the user (e.g., at enrollment). We restrict the spoofed samples involved in this study to those belonging to the reduced set of 474 (high quality) samples introduced in §V. The adversarial examples were generated from spoofed samples in this set using our $Full\ attack$ algorithm. The study also includes bonafide samples, selected from the evaluation subset of ASVspoof2019. We do not listen to adversarial samples before using them in our study. We use the same human-imperceptibility study from FakeBob [26], randomly assigning participants one of two tasks to be completed online: *Task 1: Clean or Noisy.* Participants listen to 24 samples and indicate whether each is noisy. The options are *clean*, *noisy*, or *not sure*. We randomly select 12 spoofed samples from the batch described above and 12 adversarials, crafted from spoofed recordings in the same batch (not necessarily the same samples). To ensure quality of the responses, we include three

CM \ ASV	ASVTarget1		ASVTarget2		ASVTarget3		Resemblyzer		AWS Voice ID	
	Spoofing (3 Atts./6Atts.)	*Our Attack (3 Atts./6 Atts.)*	*Spoofing (3 Atts./6Atts.)*	*Our Attack (3 Atts./6 Atts.)*	*Spoofing (3 Atts./6Atts.)*	*Our Attack (3 Atts./6 Atts.)*	*Spoofing (3 Atts./6Atts.)*	*Our Attack (3 Atts./6 Atts.)*	*Spoofing (3 Atts./6Atts.)*	*Our Attack (3 Atts./6 Atts.)*
LCNN	*1.88%/3.72%*	***34.7%/57.36%***	*3.73%/7.33%*	***31.79%/53.47%***	*9.18%/17.52%*	***39.49%/63.39%***	*2.5%/4.94%*	***17.38%/31.74%***	*1.91%/3.78%*	***4.9%/9.55%***
AIR	*1.88%/3.72%*	***84.79%/97.69%***	*3.12%/6.14%*	***78.11%/95.21%***	*6.8%/13.14%*	***87.42%/98.42%***	*2.5%/4.94%*	***71.15%/91.68%***	*1.91%/3.78%*	***60.9%/84.71%***
AIR_AM	*2.5%/4.94%*	***79.27%/95.7%***	*4.35%/8.5%*	***71.39%/91.82%***	*10.36%/19.65%*	***82.52%/96.94%***	*3.73%/7.33%*	***62.29%/85.78%***	*0.63%/1.25%*	***48.11%/73.07%***
DARTS	*1.25%/2.49%*	***39.1%/62.92%***	*3.12%/6.14%*	***35.92%/58.93%***	*10.94%/20.69%*	***43.3%/67.86%***	*1.88%/3.72%*	***23.9%/42.09%***	*1.91%/3.78%*	***6.51%/12.6%***
LCNN_LSTM	*1.25%/2.49%*	***32.62%/54.6%***	*1.88%/3.72%*	***27.92%/48.04%***	*8.59%/16.44%*	***35.11%/57.89%***	*2.5%/4.94%*	***20.21%/36.34%***	*1.91%/3.78%*	***7.57%/14.56%***
MCG	*0.0%/0.0%*	***90.07%/99.01%***	*1.25%/2.49%*	***84.16%/97.49%***	*5.56%/10.82%*	***91.83%/99.33%***	*1.88%/3.72%*	***80.21%/96.08%***	*1.91%/3.78%*	***71.63%/91.95%***
MLCG	*4.96%/9.67%*	***80.4%/96.16%***	*5.56%/10.82%*	***73.24%/92.84%***	*11.53%/21.72%*	***83.68%/97.34%***	*10.36%/19.65%*	***70.19%/91.11%***	*9.38%/17.88%*	***59.69%/83.75%***
RES2NET	*2.5%/4.94%*	***30.94%/52.31%***	*3.73%/7.33%*	***28.78%/49.28%***	*8.59%/16.44%*	***35.51%/58.42%***	*4.96%/9.67%*	***22.08%/39.29%***	*3.18%/6.25%*	***11.21%/21.17%***
SSNET	*4.35%/8.5%*	***88.22%/98.61%***	*4.35%/8.5%*	***82.52%/96.94%***	*8.0%/15.35%*	***89.72%/98.94%***	*5.56%/10.82%*	***80.58%/96.23%***	*4.43%/8.67%*	***78.83%/95.52%***
RAWGAT_ST	*3.73%/7.33%*	***24.8%/43.45%***	*4.35%/8.5%*	***19.27%/34.83%***	*9.18%/17.52%*	***27.92%/48.04%***	*7.4%/14.25%*	***16.9%/30.95%***	*0.63%/1.25%*	***9.15%/17.46%***
RAWDARTS	*0.63%/1.25%*	***33.04%/55.16%***	*1.25%/2.49%*	***30.09%/51.13%***	*4.96%/9.67%*	***35.51%/58.42%***	*2.5%/4.94%*	***19.27%/34.83%***	*2.56%/5.05%*	***8.1%/15.55%***
AASSIST	*4.35%/8.5%*	***48.36%/73.33%***	*4.35%/8.5%*	***41.03%/65.23%***	*6.8%/13.14%*	***51.45%/76.43%***	*4.96%/9.67%*	***40.65%/64.77%***	*2.56%/5.05%*	***30.21%/51.3%***
AASSIST-L	*3.12%/6.14%*	***43.67%/68.27%***	*1.88%/3.72%*	***35.92%/58.93%***	*6.8%/13.14%*	***46.6%/71.48%***	*4.96%/9.67%*	***35.92%/58.93%***	*1.28%/2.55%*	***26.66%/46.22%***
WAV2VEC	*4.35%/8.5%*	***21.62%/38.57%***	*3.12%/6.14%*	***17.86%/32.53%***	*5.56%/10.82%*	***24.35%/42.78%***	*4.96%/9.67%*	***15.44%/28.49%***	*4.43%/8.67%*	***8.62%/16.49%***

Table IV: Attack results against combined ASV-CM-STT deployments.

Frontend	CQT			Raw Waveform				
Attack \ System	***MCG***	***RES2NET***	***SSNET***	***RAWGAT_ST***	***RAWDARTS***	***AASIST***	***AASIST-L***	***WAV2VEC***
Spoofing	*10.95%*	*10.52%*	*13.8%*	*8.09%*	*11.55%*	*8.67%*	*10.53%*	*1.52%*
Our attack	*32.46%*	*30.70%*	*57.46%*	*22.15%*	*23.03%*	*26.54%*	*23.03%*	*6.03%*

Table V: Attack results against over-telephony-network CMs.

additional samples that are completely silent as a concentration test. Respondents who find these samples noisy fail the concentration test and their questionnaires are excluded.

Task 2: Identify The Speaker. Participants listen to 24 sample pairs and indicate for each pair whether the samples were uttered by the same speaker. The options are *same*, *different*, and *not sure*. We randomly select 6 speakers (3 males and 3 females) from the reduced set. For each speaker we generate four pairs: 1) *Same Benign-Benign (SBB)*—bonafide samples by the same speaker, 2) *Different Benign-Benign (DBB)*—bonafide samples from different (same gender) speakers, 3) *Same Benign-Spoofed (SBS)*—samples by the same speaker; one bonafide and one spoofed, 4) *Same Adversarial-Benign (SAB)*—samples by the same speaker; one bonafide and one adversarial. As a concentration test, we include three opposite-gender pairs of bonafide samples. Responses with answers to the concentration tasks not finding the speakers different are disqualified (as in FakeBob). To isolate the effects of our attack, adversarials in SAB pairs were all generated from the spoofed recordings in the SBS pairs. Our objective is comparing the adversarials' results to those collected for their spoofed precursors. We include the SBB and DBB pairs as a quality test (see below). Compared to FakeBob, our task adds two categories, SBS and SAB, and omits bonafide-adversarial pairs, since they generate adversarials from benign samples while we use spoofed samples.

For both tasks, we only select adversarials that successfully bypass several CMs to evaluate the attack's overall potential.

Results. We ran our IRB-approved study on MTurk [87] and received 21 responses per task. We restricted our participants to those with an MTurk approval rate of 95% or higher. Out of all responses, two (one per task) failed the concentration test, leaving us with 20 responses for each. As opposed to FakeBob, we reveal our study's purpose to participants, guaranteeing they simulate real judges inspecting suspicious samples.

Task 1. As shown in Fig. 2, 40% of our participants heard noise in spoofed samples, compared to 55% for adversarial samples. Spoofed samples are machine-generated, which may leave cues that can be spotted by a vigilant listener. This explains the relatively high rate of spoofed (baseline) samples flagged as noisy. Modern algorithms can generate better fake speech and reduce these rates [51]. However, our transformations increase this rate by only a factor of 1.375x (from 40% to 55%), which is relatively low. Compared to FakeBob, our transformations are much less obvious as their attack produced adversarial samples $\sim$3x noisier on average w.r.t their initial samples. We focus on this degradation factor, since their attack starts from cleaner (bonafide) samples, while ours modifies fake speech, which may be initially noisier. Finally, noise level is simply a quality metric, not ultimate to determine the attack's potential in fooling humans (even natural samples may include noise). *Task 2* is the more relevant metric.

Task 2. As done in FakeBob and in accordance with our description of HJ's role, we say that a listener can differentiate between the speakers of two samples only if the user selects "different" for the two samples in the pair ("not sure" still indicates uncertainty). 74% of our participants believe that samples in DBB pairs were uttered by different speakers (see Fig. 2), and only 18% deemed SBB pairs to be from the different speakers, indicating the high quality of our responses. For SAB pairs, only 50% of participants deemed the speakers different. Compared to SBS pairs, we still see a degradation as the detection probability for those is 30%, which is expected since our transformations manipulate the spectra of the samples. However, even after the degradation, detection is only as good as a random guess (50%), demonstrating the inability to spot our adversarials. In comparison, FakeBob reports results for two types of adversarials; effective and ineffective. Effective adversarials remain successful after being played over the air, while ineffective ones lose this ability. For effective adversarials, FakeBob reports that 54% of their participants could tell that they belonged to a different speaker (compared to the benign sample in the pair), while for ineffective ones, this rate drops to $\sim$30%.

Our adversarials (50% detection) outperform FakeBob's effective adversarials (54% detection), highlighting their quality. When comparing our results (50% detection) to FakeBob's ineffective samples (30% detection), it is important to note that our study is more restrictive; participants were informed

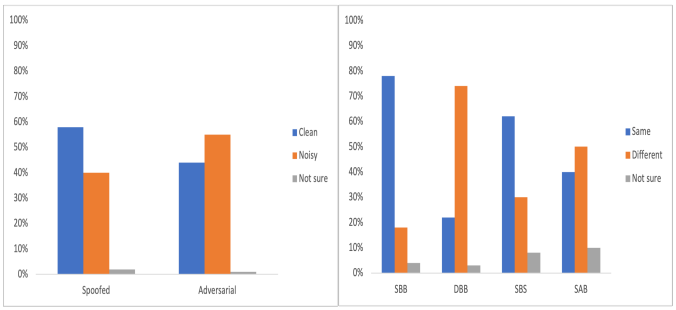

Figure 2: *Results for Tasks 1 and 2.*

that some samples are machine-generated, making them more vigilant and harder to fool [13], as opposed to FakeBob, wherein the study's purpose is never disclosed to participants. Second, FakeBob's study contained more samples per category, whereas ours has more categories but fewer samples in each, making a single "bad" frequently-spotted adversarial more influential. Considering this and our adversarials' 50% detection avoidance rate, the study proves our attack's strength.

IX. Discussion

Practicality. Our attack takes around 4 seconds per attempt and requires a limited number of attempts, making it highly practical. The attacker first needs to generate spoofed samples, which is instant with modern VC/SS algorithms [88].

Potential defenses. *Traditional adversarial defenses:* These include methods such as spatial smoothing, audio squeezing, and adversarial training [89]. FakeBob [26] shows smoothing and squeezing are ineffective against attacks on acoustic systems, and adversarial training and similar methods [90]–[94] lose effectiveness against unseen attacks [95]. We also believe our attack is naturally robust to such defenses against optimization-based adversarials since it conceptually deviates from the hypotheses behind them.

Liveness detection: As explained in §II-B, *active* liveness detection methods are hyper-sensitive and therefore less practical. Other non-traditional approaches include Voicefox [96], which suggests that higher mistranscription rates by STTs may indicate that a sample is spoofed but the quality of modern spoofing algorithms and the pace at which they develop will soon make such methods invalid. Blue et al. [85] detect spoofed speech through vocal tract reconstruction. However, vocal tract manipulation tools [60] may defeat this defense.

Attack-specific defenses: We considered defenses specific for our attack. The first is eliminating silence (applying VAD), nullifying the effects of "non-speech" cues that we inject in F_1&F_2. Although we state that these intervals are crucial for CMs (see §IV-B), Zhang et al. [97] propose a model ($SENet\_FFT$) capable of classifying silence-free speech. We implement it but find applying VAD leading to poor performance on the test set even without our attack (EER=24.34%). We attribute this to us using a well-known VAD algorithm ($WebRTCVad$ [62]), which is aggressive at filtering out silence, while they provide no details regarding their probably much more permissive VAD algorithm, which leaves many silence intervals in the samples, enabling proper functionality. Thus, VAD is not a viable defense. Noise reduction will similarly filter out vital silence information.

The second defense is low-pass filtering. We boost the center of the spectrum, but higher components are amplified as well. These should not be considered for audio classification as they are noise-dominated, unreliable and missing in bandwidth-limited scenarios (see §VII). Yet, this byproduct could be a key reason behind the attack's success, which can be defeated by rejecting information in this range. Zhang et al. [97] also suggest using $SENet\_FFT$ with frequencies above $4KHz$ discarded to increase robustness. We reproduce this, and achieve a comparable EER—2.11% (without VAD). Afterward, we run our attack. For spoofed samples, the model is robust, with a success rate of 0.63%. Yet, with our attack, the rate rises to 11.18% (single attempt), proving our spectral manipulation effects (F_3&F_5) are deeply rooted in the central regions of the spectrum, where relevant information resides.

Identifying promising VA improvements: The best evaluated defense against our attack comes in the form of a CM already evaluated (WAV2VEC), for it having the potential to reject background noise and high frequency components. We explain this in §VI-A, but demonstrate how even this defense falls short. The over-telephony setting, however, reveals additional interesting findings: transcoding together with the bandwidth loss, combined with training on noise-augmented data (e.g., WAV2VEC) can lead to a degradation in the attack's success (see §VII). We believe that this is a promising research avenue to explore in the search for robust spoofing countermeasures, both in the designated-app setting and for phone-call authentication. Despite the fact that the loss incurred by transcoding and down-sampling is a byproduct of transmission over the telephony network, these components can be simulated algorithmically, potentially raising the bar for attackers who are currently more easily capable of mounting attacks in the designated-app scenario, especially when combined with channel invariant models (obtained by training on noise-augmented data) such as WAV2VEC. However, at this embryonic stage our attack still achieves concerning success rates after a number of attempts against this configuration (see §VII).

Outlook. Spoofing detection is a cat-and-mouse game, wherein novel algorithms and threats are emerging at an unprecedented rate without a reliable solution to make for robust VA. The threat of replay attacks alone is still unaccounted for, despite many works describing potential solutions in top venues [44], [98], as new studies are constantly demonstrating how slightly different settings or audio channels [13], [99] and attacks [98] can severely degrade the performance of such systems. The facts are similar for fake speech detection, as despite the tremendous efforts invested so far, reports discussing the lack of generalization to different settings and algorithms continue to appear [54], [97], [100]. Nonetheless, replay/spoofing detection systems continue to gain popularity in practice due to the lack of better alternatives. While recent efforts present somewhat generalizeable models (e.g., WAV2VEC), the true ability of these to generalize to all settings remains an open

question. Regardless, we attacked the state-of-the-art available *passive* solutions, which are suitable and being deployed in security-critical environments, and proved their vulnerability to our transformations under the settings and assumptions at which they perform best. In the future, it is imperative to study the link between our findings and generalization.

X. Related Work

Non-Proactive Spoofing Attacks: These attacks fall under four sub-classes: replay [42], mimicking [40], voice conversion (VC) [41], and speech synthesis (SS) [10] (see §II). Replay and mimicking are often disqualified as ASVs can be easily made robust against them (see §II). We restrict the discussion to SS/VC attacks, which are successful in fooling ASVs, achieving success rates up to 100% [13]. This motivated the development of (CMs) [44], [101] to protect ASVs. Many advanced CMs have since appeared, demonstrating robustness and generalization [14], [19], [44], [76], [101], and such CMs are deployed by leading VA vendors [18]. Thus, we disqualify non-proactive spoofing attacks assuming CMs defeat them.

Adversarial Attacks on ASVs and STTs: In targeted adversarial attacks on ASVs, the attacker starts from some audio sample (typically, but not necessarily a human recording) and adversarially optimizes it to fool the ASV into believing it was uttered by the victim. Against STTs, the task is to make them transcribe the sample into a text of the attacker's choice.

The literature offers many such attacks. However, they fail to break VA under security-critical assumptions. First, many assume (semi-)whitebox access (at least some crucial parameters are known) to the target models [21], [79], [102] (STTs), [22], [39], [103]–[107] (ASVs), and [20] (both). They all either fail to attack blackbox systems or rely on transferability to do so. However, adversarial transferability across ASVs/STTs has been proven very limited [86], and works demonstrating good transferability rates (for targeted attacks) only do so when their targets and shadows share many commonalities, making them unsuitable for our black-box setting. Other approaches attacking STTs [23]–[25] and ASVs [25]–[27] circumvent the lack of transferability via target-specific attacks that repeatedly query the target to infer its decision boundary. These are query-inefficient and do not fit into our threat model. Recently, NI-OCAAM [25] satisfied the large query budget requirement. Yet, their techniques are based on CommanderSong [79], which is extremely slow, violating the response time constraint (see §II). The same issue is faced by SirenAttack [28].

Another class of attacks operates at the signal processing level, such as Pipe Overflow [100], which fools ASVs via acoustic resonance perturbations. However, it assumes access to the probability scores of the target system, requires external devices, and is not query efficient. Abdullah et al. [29] target ASVs/STTs and show high success rates. Yet, it is untargeted and can only make a model arbitrarily misclassify, which is of no use in our case. Abdullah et al. [108] show how certain transformations can make audio impossible to understand by humans, while maintaining correct classification by ASVs (STTs). However, they consider a different goal, which is hiding a *previously-recorded* bonafide sample of the victim in unintelligible audio to deceive humans, while we do not assume access to such recordings, as the phrases to repeat are generated randomly by the target (see §II).

Adversarial examples against ASVs/STTs can probably fool CMs. CMs detect machine-generated speech, while the above adversarials typically introduce changes to *natural* speech, leaving them undetectable by CMs. In fact, Ahmed et al. [100] conduct an experiment proving their attack bypasses CMs. It is crucial to understand that CMs may not defend against these attacks, but are still highly successful against spoofing [19], [44], making them essential in critical environments.

Adversarial Attacks on CMs: Adversarial attacks on CMs are advanced attacks that combine spoofing with adversarial examples and are conceptually similar to our own. They add adversarial perturbations to SS/VC outputs to get CMs to misclassify them as bonafide. Due to the vulnerability of ASVs to spoofing and the high quality of spoofing algorithms, such perturbations, when subtle enough, can make spoofed speech bypass all VA components combined. However, similar to adversarial attacks on ASVs/STTs, known attacks on CMs still have not achieved this goal under security-critical assumptions.

Most attacks on CMs (or joint ASV-CM systems) [30]–[33] have been whitebox and are of limited applicability, or relied on transferability to attack blackbox systems. However, transferability attacks against CMs, as is the case with ASVs/STTs, are extremely limited in potential and we attribute observed high transferability rates to not considering fully blackbox settings and experimenting with small sets of CMs lacking robustness in real-world settings (see §VI-A). Finally, Ding et al. [109], [110] propose a novel method for adversarially enhancing VC outputs to bypass CMs. Yet, their attack is only (semi-)whitebox and they experiment with a single CM. There are target-specific attacks that require many target queries [34], [111] to estimate the decision boundaries. These attacks are ineffective under security-critical constraints (see §II). In a concurrent work, Hua et al. [112] propose enhancing VC outputs to spoof CMs by adding global noise and replacing the silent intervals in the generated speech with ones from bonafide samples in a post-generation phase. We integrate similar (yet different) stages into our transformations (see §IV-B). However, the improvement of these alone is not drastic, and other steps are vital to generate robust adversarials.

XI. Conclusion

We presented the first practical attack on security-critical VA, through targeting a newly-identified weakest link— CMs. Our novel targeted, real-time and model-agnostic attack generates high-quality fake speech fooling both machines and humans, and enabling the attacker to impersonate the victim, severely compromising users' security. Our attack's success is concerning, primarily due to it being mounted in blackbox and practical settings, including the designated-app and phone conversation scenarios. Our findings highlight the severe pitfalls of modern VA systems and the need for defenses.

ACKNOWLEDGMENTS

We gratefully acknowledge the support of the Waterloo-Huawei Joint Innovation Laboratory for funding this research, and the Compute Canada Foundation (CCF) for their resources that made our experiments possible.

REFERENCES

[1] The Insight Partners. Voice Biometrics Market Size Worth $4.82Bn, Globally, by 2028. https://www.prnewswire.com/news-releases/voice-biometrics-market-size-worth-4-82bn-globally-by-2028-at-20-6-cagr---exclusive-report-by-the-insight-partners-301552781.html.

[2] Forbes. Citi Uses Voice Prints To Authenticate Customers Quickly And Effortlessly. hhttps://www.forbes.com/sites/tomgroenfeldt/2016/06/27/citi-uses-voice-prints-to-authenticate-customers-quickly-and-effortlessly/?sh=5abfb980109c.

[3] First Direct. What is Voice ID security? https://www1.firstdirect.com/banking/ways-to-bank/telephone-banking/#voice-id-security.

[4] Nuance. Nuance VocalPassword : voice biometrics authentication. https://www.nuance.com/content/dam/nuance/en_us/collateral/enterprise/brochure/br-vocalpassword-en-us.pdf.

[5] Aculab. https://www.aculab.com.

[6] Nuance. ING Introduces a Voice-Controlled Mobile Banking App Powered by Nuance. https://news.nuance.com/2014-09-16-ING-Introduces-a-Voice-Controlled-Mobile-Banking-App-Powered-by-Nuance.

[7] ——. Nuance Voice Biometrics Quickly Identifies Customers and Strengthens Security for National Australia Bank. https://news.nuance.com/2020-05-27-Nuance-Voice-Biometrics-Quickly-Identifies-Customers-and-Strengthens-Security-for-National-Australia-Bank.

[8] TD Bank. Voice Print System. https://www.td.com/ca/products-services/investing/td-direct-investing/trading-platforms/voice-print-system-enroll.jsp.

[9] VICE. (2023, Feb) How I broke into a bank account with an AI-generated voice. https://www.vice.com/en/article/dy7axa/how-i-broke-into-a-bank-account-with-an-ai-generated-voice.

[10] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions," in *IEEE ICASSP 2018*.

[11] T. Toda, A. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 2222–2235, 12 2007.

[12] D. Mukhopadhyay, M. Shirvanian, and N. Saxena, "All your voices are belong to us: Stealing voices to fool humans and machines," in *ESORICS 2015*.

[13] E. Wenger, M. Bronckers, C. Cianfarani, J. Cryan, A. Sha, H. Zheng, and B. Y. Zhao, ""Hello, It's Me": Deep learning-based speech synthesis attacks in the real world," in *ACM CCS 2021*.

[14] J. Yamagishi, X. Wang, M. Todisco, M. Sahidullah, J. Patino, A. Nautsch, X. Liu, K. A. Lee, T. Kinnunen, N. Evans *et al.*, "Asvspoof 2021: accelerating progress in spoofed and deepfake speech detection," in *ASVspoof 2021 Workshop*.

[15] O. Research. 2022 Intelligent Authentication and Fraud Prevention Intelliview. https://www.nuance.com/content/dam/nuance/en_us/collateral/enterprise/report/ar-opus-2022-intelligent-authentication-and-fraud-prevention-intelliview-en-us.pdf.

[16] Nuance. Gatekeeper: Cloud-native biometric security for every channel. https://www.nuance.com/omni-channel-customer-engagement/authentication-and-fraud-prevention/gatekeeper.html.

[17] ID R&D. ID R&D Voice. https://www.idrnd.ai/voice-anti-spoofing.

[18] ——. ID R&D ranks first in detecting synthetic speech in Global ASVspoof Challenge. https://www.idrnd.ai/id-rd-ranks-first-in-detecting-synthetic-speech-in-global-asvspoof-challenge/.

[19] H. Tak, M. Todisco, X. Wang, J.-w. Jung, J. Yamagishi, and N. Evans, "Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation," *arXiv preprint arXiv:2202.12233*, 2022.

[20] Z. Li, Y. Wu, J. Liu, Y. Chen, and B. Yuan, "AdvPulse: Universal, synchronization-free, and targeted audio adversarial attacks via subsecond perturbations," in *ACM CCS 2020*.

[21] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*.

[22] F. Kreuk, Y. Adi, M. Cisse, and J. Keshet, "Fooling end-to-end speaker verification with adversarial examples," in *IEEE ICASSP 2018*.

[23] M. Alzantot, B. Balaji, and M. Srivastava, "Did you hear that? Adversarial examples against automatic speech recognition," in *NIPS 2018*.

[24] Y. Chen, X. Yuan, J. Zhang, Y. Zhao, S. Zhang, K. Chen, and X. Wang, "Devil's whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices," in *USENIX Security 2020*. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/chen-yuxuan

[25] B. Zheng, P. Jiang, Q. Wang, Q. Li, C. Shen, C. Wang, Y. Ge, Q. Teng, and S. Zhang, "Black-box adversarial attacks on commercial speech platforms with minimal information," in *ACM CCS 2021*.

[26] G. Chen, S. Chen, L. Fan, X. Du, Z. Zhao, F. Song, and Y. Liu, "Who is real Bob? Adversarial attacks on speaker recognition systems," in *IEEE S&P, Oakland 2019*.

[27] H. Luo, Y. Shen, F. Lin, and G. Xu, "Spoofing speaker verification system by adversarial examples leveraging the generalized speaker difference," *Security and Communication Networks*, vol. 2021.

[28] T. Du, S. Ji, J. Li, Q. Gu, T. Wang, and R. Beyah, "SirenAttack: Generating adversarial audio for end-to-end acoustic systems," in *ACM ASIACCS 2020*.

[29] H. Abdullah, M. Rahman, W. Garcia, K. Warren, A. S. Yadav, T. Shrimpton, and P. Traynor, "Hear "no evil", see "Kenansville": Efficient and transferable black-box attacks on speech recognition and voice identification systems," in *IEEE S&P, Oakland 2021*.

[30] S. Liu, H. Wu, H.-y. Lee, and H. Meng, "Adversarial attacks on spoofing countermeasures of automatic speaker verification," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019.

[31] Y. Zhang, Z. Jiang, J. Villalba, and N. Dehak, "Black-box attacks on spoofing countermeasures using transferability of adversarial examples," in *Interspeech 2020*.

[32] X. Zhang, X. Zhang, X. Zou, H. Liu, and M. Sun, "Towards generating adversarial examples on combined systems of automatic speaker verification and spoofing countermeasure," *Security and Communication Networks*, 2022.

[33] X. Zhang, X. Zhang, W. Liu, X. Zou, M. Sun, and J. Zhao, "Waveform level adversarial example generation for joint attacks against both automatic speaker verification and spoofing countermeasures," *Engineering Applications of Artificial Intelligence*, vol. 116, p. 105469, 2022.

[34] A. Gomez-Alanis, J. A. Gonzalez-Lopez, and A. M. Peinado, "Adversarial transformation of spoofing attacks for voice biometrics," *arXiv preprint arXiv:2201.01226*, 2022.

[35] Amazon. Connect VoiceID. https://aws.amazon.com/connect/voice-id/.

[36] Google. Cloud STT. https://cloud.google.com/speech-to-text.

[37] Microsoft. Azure STT. https://azure.microsoft.com/en-us/products/cognitive-services/speech-to-text/.

[38] T. F. Zheng and L. Li, "Robustness-related issues in speaker recognition," *SpringerBriefs in Signal Processing*, 2017.

[39] X. Li, J. Zhong, X. Wu, J. Yu, X. Liu, and H. Meng, "Adversarial attacks on GMM i-vector based speaker verification systems," in *IEEE ICASSP 2020*.

[40] Y. W. Lau, M. Wagner, and D. Tran, "Vulnerability of speaker verification to voice mimicking," in *International Symposium on Intelligent Multimedia, Video and Speech Processing*, 2004.

[41] T. Kinnunen, J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, and Z. Ling, "A spoofing benchmark for the 2018 voice conversion challenge: Leveraging from spoofing countermeasures for speech artifact assessment," *arXiv preprint arXiv:1804.08438*, 2018.

[42] Z. Wu and H. Li, "On the study of replay and voice conversion attacks to text-dependent speaker verification," *Multimedia Tools and Applications*, vol. 75, 05 2016.

[43] R. K. Das, X. Tian, T. Kinnunen, and H. Li, "The attacker's perspective on automatic speaker verification: An overview," in *Interspeech 2020*.

[44] M. E. Ahmed, I.-Y. Kwak, J. H. Huh, I. Kim, T. Oh, and H. Kim, "Void: A fast and light voice liveness detection system," in *USENIX Security 2020*.

[45] Y. Meng, J. Li, M. Pillari, A. Deopujari, L. Brennan, H. Shamsie, H. Zhu, and Y. Tian, "Your microphone array retains your identity: A

robust voice liveness detection system for smart speaker," in *USENIX Security*, 2022.
[46] L. Zhang, S. Tan, J. Yang, and Y. Chen, "VoiceLive: A phoneme localization based liveness detection for voice authentication on smartphones," in *ACM CCS 2016*.
[47] L. Zhang, S. Tan, and J. Yang, "Hearing your voice is not enough: An articulatory gesture based liveness detection for voice authentication," in *ACM CCS 2017*.
[48] S. A. Anand, J. Liu, C. Wang, M. Shirvanian, N. Saxena, and Y. Chen, "EchoVib: Exploring voice authentication via unique non-linear vibrations of short replayed speech," in *ACM ASIACCS 2021*.
[49] Android-Rootkit. A rootkit for Android. Based on "Android platform based linux kernel rootkit" from Phrack Issue 68. https://github.com/hiteshd/Android-Rootkit.
[50] Cinta Infinita. Android: How to Bypass Root Check and Certificate Pinning. https://medium.com/@cintainfinita/android-how-to-bypass-root-check-and-certificate-pinning-36f74842d3be.
[51] H. Zen, R. Clark, R. J. Weiss, V. Dang, Y. Jia, Y. Wu, Y. Zhang, and Z. Chen, "LibriTTS: A corpus derived from LibriSpeech for text-to-speech," in *Interspeech 2019*. [Online]. Available: https://arxiv.org/abs/1904.02882
[52] ANZ. Voice ID in the ANZ App. https://www.anz.com.au/ways-to-bank/mobile-banking-apps/voice-id/.
[53] Datafloq. Social Engineering Attacks by the Numbers: Prevalence, Costs, and Impact. https://datafloq.com/read/social-engineering-attacks-numbers-cost/6068#:~:text=CyberEdge%20reports%20that%20the%20number,76%20percent%20a%20year%20later.
[54] X. Liu, X. Wang, M. Sahidullah, J. Patino, H. Delgado, T. Kinnunen, M. Todisco, J. Yamagishi, N. Evans, A. Nautsch, and K. A. Lee, "Asvspoof 2021: Towards spoofed and deepfake speech detection in the wild," *arXiv preprint arXiv:2210.02437*, 2022.
[55] J. Frank and L. Schönherr, "Wavefake: a data set to facilitate audio deepfake detection," in *NeurIPS 2021*.
[56] M. Todisco, H. Delgado, and N. W. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients." in *Odyssey*, 2016.
[57] N. R. French and J. C. Steinberg, "Factors governing the intelligibility of speech sounds," *The Journal of the Acoustical Society of America*, vol. 19, no. 1, pp. 90–119, 1947.
[58] Hydrogenaudio. Pre-emphasis. https://wiki.hydrogenaud.io/index.php?title=Pre-emphasis.
[59] T. Sainburg, "timsainb/noisereduce: v1.0," Jun. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3243139
[60] Praat. Vocal Toolkit. https://www.praatvocaltoolkit.com/index.html.
[61] B. McFee, C. Raffel, D. Liang, D. Ellis, M. Mcvicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *SciPy 2015*.
[62] Py-webrtcvad. Python interface to the WebRTC Voice Activity Detector (VAD). https://github.com/wiseman/py-webrtcvad.
[63] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.
[64] Amazon. Amazon Connect. https://aws.amazon.com/connect.
[65] Twilio. https://www.twilio.com/.
[66] J.-w. Jung, H.-S. Heo, H. Tak, H.-j. Shim, J. S. Chung, B.-J. Lee, H.-J. Yu, and N. Evans, "AASIST: Audio anti-spoofing using integrated spectro-temporal graph attention networks," in *arXiv preprint arXiv:2110.01200*, 2021.
[67] H. Tak, J.-w. Jung, J. Patino, M. Kamble, M. Todisco, and N. Evans, "End-to-end spectro-temporal graph attention networks for speaker verification anti-spoofing and speech deepfake detection," in *ASVspoof 2021 Workshop*.
[68] G. Hua, A. Bengjinteoh, and H. Zhang, "Towards end-to-end synthetic speech detection," *IEEE Signal Processing Letters*, 2021.
[69] W. Ge, J. Patino, M. Todisco, and N. Evans, "Raw Differentiable Architecture Search for Speech Deepfake and Spoofing Detection," in *2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, 2021.
[70] X. Li, X. Wu, H. Lu, X. Liu, and H. Meng, "Channel-wise gated res2net: Towards robust detection of synthetic speech attacks," in *Interspeech 2021*.
[71] X. Wang and J. Yamagishi, "A Comparative Study on Recent Neural Spoofing Countermeasures for Synthetic Speech Detection," in *Interspeech 2021*.
[72] Y. Zhang, F. Jiang, and Z. Duan, "One-class learning towards synthetic voice spoofing detection," *IEEE Signal Processing Letters*, vol. 28, pp. 937–941, 2021.
[73] X. Li, N. Li, C. Weng, X. Liu, D. Su, D. Yu, and H. Meng, "Replay and synthetic speech detection with res2net architecture," in *IEEE ICASSP 2021*.
[74] W. Ge, M. Panariello, J. Patino, M. Todisco, and N. Evans, "Partially-connected differentiable architecture search for deepfake and spoofing detection," in *Interspeech 2021*.
[75] G. Lavrentyeva, S. Novoselov, A. Tseren, M. Volkova, A. Gorlanov, and A. Kozlov, "STC antispoofing systems for the ASVspoof2019 challenge," in *Interspeech 2019*.
[76] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. Kinnunen, and K. A. Lee, "ASVspoof 2019: Future horizons in spoofed and fake audio detection," in *Interspeech 2019*.
[77] VentureBeat. Google releases dataset to help AI systems spot fake audio recordings. https://venturebeat.com/ai/google-releases-dataset-to-help-ai-systems-determine-if-an-audio-recording-is-real/.
[78] Kaldi ASR. https://github.com/kaldi-asr/kaldi.
[79] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, "CommanderSong: A systematic approach for practical adversarial voice recognition," in *USENIX Security 2018*.
[80] Resemblyzer. https://github.com/resemble-ai/Resemblyzer.
[81] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *IEEE ICASSP 2018*.
[82] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee *et al.*, "ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Computer Speech & Language*, vol. 64, p. 101114, 2020.
[83] VoxCeleb. https://www.robots.ox.ac.uk/~vgg/data/voxceleb/vox1.html.
[84] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," *arXiv:1510.08484v1*, 2015.
[85] L. Blue, K. Warren, H. Abdullah, C. Gibson, L. Vargas, J. O'Dell, K. Butler, and P. Traynor, "Who are you (I really wanna know)? Detecting audio deepfakes through vocal tract reconstruction," in *USENIX Security 2022*.
[86] H. Abdullah, K. Warren, V. Bindschaedler, N. Papernot, and P. Traynor, "The faults in our ASRs: An overview of attacks against automatic speech recognition and speaker identification systems," in *IEEE S&P, Oakland 2020*.
[87] Amazon. Mechanical Turk Platform. https://www.mturk.com/.
[88] Nuance. Text-To-Speech. https://www.nuance.com/omni-channel-customer-engagement/voice-and-ivr/text-to-speech.html.
[89] H. Wu, S. Liu, H. Meng, and H.-y. Lee, "Defense against adversarial attacks on spoofing countermeasures of asv," in *IEEE ICASSP 2020*.
[90] M. Pal, A. Jati, R. Peri, C.-C. Hsu, W. AbdAlmageed, and S. Narayanan, "Adversarial defense for deep speaker recognition using hybrid adversarial training," in *IEEE ICASSP 2020*.
[91] S. Joshi, J. Villalba, P. Żelasko, L. Moro-Velázquez, and N. Dehak, "Adversarial attacks and defenses for speaker identification systems," *arXiv preprint arXiv:2101.08909*, 2021.
[92] Q. Wang, P. Guo, S. Sun, L. Xie, and J. H. Hansen, "Adversarial regularization for end-to-end robust speaker verification." in *Interspeech*, 2019, pp. 4010–4014.
[93] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.
[94] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *arXiv preprint arXiv:1312.6114*, 2013.
[95] X. Li, N. Li, J. Zhong, X. Wu, X. Liu, D. Su, D. Yu, and H. Meng, "Investigating robustness of adversarial samples detection for automatic speaker verification," *arXiv preprint arXiv:2006.06186*, 2020.
[96] M. Shirvanian, M. Mohammed, N. Saxena, and S. A. Anand, "Voicefox: Leveraging inbuilt transcription to enhance the security of machine-human speaker verification against voice synthesis attacks," in *ACSAC 2020*.
[97] Y. Zhang, W. Wang, and P. Zhang, "The Effect of Silence and Dual-Band Fusion in Anti-Spoofing System," in *Interspeech 2021*.
[98] S. Wang, J. Cao, X. He, K. Sun, and Q. Li, "When the differences in frequency domain are compensated: Understanding and defeating

modulated replay attacks on automatic speech recognition," in *ACM CCS 2020*.
[99] H. A. Patil and M. R. Kamble, "A survey on replay attack detection for automatic speaker verification (ASV) system," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2018.
[100] S. Ahmed, Y. Wani, A. S. Shamsabadi, M. Yaghini, I. Shumailov, N. Papernot, and K. Fawaz, "Pipe Overflow: Smashing voice authentication for fun and profit," *arXiv preprint arXiv:2202.02751*, 2022.
[101] Z. Wu, J. Yamagishi, T. Kinnunen, C. Hanilçi, M. Sahidullah, A. Sizov, N. Evans, M. Todisco, and H. Delgado, "ASVspoof: the automatic speaker verification spoofing and countermeasures challenge," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 4, pp. 588–604, 2017.
[102] M. M. Cisse, Y. Adi, N. Neverova, and J. Keshet, "Houdini: Fooling deep structured visual and speech recognition models with adversarial examples," in *NIPS 2017*.
[103] Y. Xie, C. Shi, Z. Li, J. Liu, Y. Chen, and B. Yuan, "Real-time, universal, and robust adversarial attacks against speaker recognition systems," in *IEEE ICASSP 2020*.
[104] Z. Li, C. Shi, Y. Xie, J. Liu, B. Yuan, and Y. Chen, "Practical adversarial attacks against speaker recognition systems," in *21st International Workshop on Mobile Computing Systems and Applications*, 2020.
[105] Q. Wang, P. Guo, and L. Xie, "Inaudible adversarial perturbations for targeted attack in speaker recognition," in *Interspeech 2020*.
[106] J. Villalba, Y. Zhang, and N. Dehak, "x-vectors meet adversarial attacks: Benchmarking adversarial robustness in speaker verification," in *Interspeech 2020*.
[107] A. Jati, C.-C. Hsu, M. Pal, R. Peri, W. AbdAlmageed, and S. Narayanan, "Adversarial attack and defense strategies for deep speaker recognition systems," *Computer Speech & Language*, vol. 68, p. 101199, 2021.
[108] H. Abdullah, W. Garcia, C. Peeters, P. Traynor, K. R. Butler, and J. Wilson, "Practical hidden voice attacks against speech and speaker recognition systems," in *NDSS 2019*.
[109] Y.-Y. Ding, H.-J. Lin, L.-J. Liu, Z.-H. Ling, and Y. Hu, "Robustness of speech spoofing detectors against adversarial post-processing of voice conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3415–3426, 2021.
[110] Y.-Y. Ding, J.-X. Zhang, L.-J. Liu, Y. Jiang, Y. Hu, and Z.-H. Ling, "Adversarial post-processing of voice conversion against spoofing detection," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2020.
[111] Y. Jiang and D. Ye, "Black-box adversarial attacks against audio forensics models," *Security and Communication Networks*, 2022.
[112] H. Hua, Z. Chen, Y. Zhang, M. Li, and P. Zhang, "Improving spoofing capability for end-to-end any-to-many voice conversion," in *1st International Workshop on Deepfake Detection for Audio Multimedia*, 2022.
[113] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE S&P, Oakland 2017*.
[114] Rev. Microsoft Azure Speech Recognition vs. Rev AI Speech to Text API. https://www.rev.com/blog/resources/microsoft-azure-speech-recognition-vs-rev-ai-speech-to-text-api.

APPENDIX

A. ASV Models

Details of our chosen ASVs are in Table VI. To use Amazon Connect Voice ID and Resemblyzer, we had to calibrate them to obtain decision thresholds for our specific use case. To calibrate Voice ID, first, we enrolled all the speakers in the evaluation subset of the ASVspoof2019 dataset using the process outlined in §V. Afterward, we selected 480 **bonafide** samples from the evaluation subset of ASVspoof2019 s.t none of these were used for speaker enrollment. We denote this set as AWSCalibrate. For 240 of the samples, we invoked Voice ID with the ID of the actual speaker of the sample. For the other 240 samples, Voice ID was invoked with a different speaker's ID. The scores were used to calculate the EER threshold.

As for Resemblyzer, for each **bonafide** sample in the evaluation subset of ASVspoof2019, we performed verification with two bonafide reference samples; one that belongs to the same speaker, and another which belongs to a different speaker. The scores were used to calculate the threshold.

B. CM Calibration for The Over-Telephony Setting

Due to the costly nature of our over-telephony experiments, preventing us from transmitting thousands of samples over the phone, and the need to have well-substantiated results that warrant including the different CMs in this experiment, calibration was done in two phases. First, we use the ASVspoof2021 dataset [14], which evaluates the ability to deal with the channel effect. The results are in Table VII (first row). For reference, we include the EER (9.5%) for the baseline model in the ASVspoof2021 challenge ($RAWNET2$). The state-of-the-art WAV2VEC [19] is missing from the table, but as explained in §V, it is superior to all known models with a 0.8% EER. Performance is degraded significantly for some models due to the channel effect, while for others, the degradation is not as drastic. We filter-out models lacking generalization to the phone setting, leaving only those with EERs below 15%. While this is not practical, it is on par with the $RAWNET2$ baseline. The attacker also needs to bypass other components as well, further decreasing the actual success rates.

After choosing the models that are somewhat generalizable to the phone scenario, we empirically verify their ability to perform classification under our specific settings (i.e., when transmitting the samples using Twilio to Amazon Connect) as the different codecs used and system specifications may reveal that not all models are in fact applicable for this use case. We choose a subset, named AWSCalibrateCM, of spoofed and bonafide samples (the same 474 spoofed samples from §V and 230 other bonafide samples) from the evaluation section of ASVspoof2019 and transmit them using Twilio to our Amazon Connect instance. We then extract the transmitted samples from the conversations we record at the receiver and use these samples to test whether the CMs can still differentiate between the bonafide and spoofed samples and calculate their alternative EERs (w.r.t AWSCalibrateCM) accordingly. The results can be found in the second row of Table VII and prove the model's ability to generalize, achieving relatively low EERs. Furthermore, the thresholds for achieving these EERs (in parentheses) are higher, making the attacker's task ever harder (a higher score is needed to accept the sample). Hence, we use these EERs for our over-telephony experiment.

C. Shattering the Myth of Transferability

Our guiding principle is this: models for audio-related tasks must be robust to various types of naturally-occurring noise, the most common of which being additive (white). This is known to the anti-spoofing community [14]. Models lacking this robustness are untrustworthy and should not be deployed in real-world environments. Specifically, spoofed samples tend

System	***ShadowASV***	***ASVTarget1***	***ASVTarget2***	***ASVTarget3***	***Google/Resemblyzer***	***AWS-Voice ID***
Architecture	*GMM-IVECTOR (Gaussians: 1024, IVECTOR Dim.: 300)*	*GMM-IVECTOR (Gaussians: 2048, IVECTOR Dim.: 400)*	*GMM-IVECTOR (Gaussians: 2048, IVECTOR Dim.: 400)*	*XVECTOR*	*DNN*	*N/A*
Frontend	*MFCC*	*MFCC*	*LPMS*	*MFCC*	*Mel Spectrogram*	*N/A*
Training Set	*ASVspoof2019-dev(bonafide only)*	*ASVspoof2019-train (bonafide only)*	*ASVspoof2019-train (bonafide only)*	*Voxceleb1-Train*	*Voxceleb1&2&LibriSpeech*	*N/A*
Calibration/Eval Set	*ASVspoof2019-eval (bonafide only)*	*ASVspoof2019-eval (bonafide only)*	*ASVSpoof2019-eval (bonafide only)*	*Voxceleb1-Test*	*ASVspoof2019-eval (bonafide only)*	*AWSCalibrate*
EER	2.76%	3%	4.65%	6.421%	13%	6.45%

Table VI: Architectures, datasets, and EERs of the different ASVs w.r.t their evaluation/calibration datasets.

Frontend	***LPMS***	***LFCC***				***CQT***			***Raw Waveform***					
System / *Dataset*	***LCNN***	***AIR***	***AIR_AM***	***DARTS***	***LCNN_LSTM***	***MCG***	***MLCG***	***RES2NET***	***SSNET***	***RAWGAT_ST***	***RAWDARTS***	***AASIST***	***AASIST-L***	***RAWNET2 (BASELINE)***
ASVspoof2021	*29.37% (1.561)*	*18.231% (-0.87)*	*19.93% (0.21)*	*17.157% (2.94)*	*17.07% (0.00)*	***11.81% (0.57)***	*18.16% (0.12)*	***12.7% (0.09)***	***13.8% (1.15)***	***8.09% (-2.47)***	***11.55% (0.01)***	***8.67% (-2.7)***	***10.53% (-0.45)***	***9.5% (N/A)***
AWSCalibrateCM	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*10.95% (0.98)*	*N/A*	*10.52% (0.31)*	*10.41% (2.06)*	*4.77% (-1.35)*	*6.61% (0.00)*	*5.74% (-1.44)*	*4.33% (-0.45)*	*N/A*

Table VII: CMs' EERs (thresholds) in the phone setting w.r.t ASVspoof2021/AWSCalibrateCM. EERs below 15% are in bold.

to be clean as the outputs of computer algorithms, whereas bonafide samples are recorded using microphones by humans and are more likely to exhibit natural noise (see §IV-B). Transferability is useless unless it affects noise-robust systems.

There are currently a variety of models that have demonstrated robustness to noisy samples, making them suitable for real-world environments, such as Void [44] and WAV2VEC [19] (which was trained on noise-augmented data and with which we experiment in §VI-A). This is trivially expected, especially from commercial systems used in practice. The participating systems in ASVspoof2019, being on par with industry standards (see §V), may very well be able to generalize to noisy settings. Yet, there is no guarantee that **all** these systems do as they have not been evaluated in these settings. Previous works demonstrating adversarial attacks on CMs [30]–[33] overlook this fact, which we assume to be the main reason behind their high transferability rates. Our hypothesis is that models that exhibit a significantly degraded performance under transferability are not robust to noise, enabling **any** noise to bring them out of balance and cause them to misclassify. Additionally, such models are all overfitted to the training data and therefore it would be natural that they would share many common decision boundaries, as opposed to robust models. Our goal is to determine which systems are not robust to additive noise and test its connection to transferability.

Our large-scale experiment is this: first, we introduce additive noise to all the samples in our evaluation dataset with various budgets. Then, we generate adversarial examples using common methods (I-FGSM [63], PGD [63], Carlini & Wagner [113]) and $LCNN$ (trained on the development subset of ASVspoof2019, as opposed to the $LCNN$ target in all of our tables, which was trained on the training subset— to enforce blackbox settings) as the shadow. This system was used in previous works, leading to high transferability [30].

Additive noise and transferability results are in Tables VIII and IX, respectively, and prove a direct link between lacking noise robustness transferability. $LCNN$ shares its architecture with the shadow, so the high success rates are expectable (not blackbox). $LCNN\_LSTM$, $RES2NET$ exhibit good noise robustness and, while some transferability occurs for these, it remains limited. Systems that are highly noise-robust ($RAWGAT\_ST$, $RAWDARTS$, $AASIST$, $AASIST$-L) are insensitive to transferability. The systems vulnerable to transferability (AIR, AIR_AM, $DARTS$, MCG, $MLCG$) are those lacking noise robustness and not suitable for practical use, proving transferability is not effective in practice.

D. Results for Individual Transformations

Individual results are in Table X. We reiterate the statement made in §VI-A noting that each transformation targets a specific set of heuristics and the dependence of the CMs on these principles varies according to their design and underlying features. Hence, the transformations individually do not always succeed in attacking all CMs. The universal nature of our full attack lies in its ability to target multiple identifiable and removable cues simultaneously. $F_1 - F_7$, when employed together as shown in §VI-A, suffice to achieve this goal. Individual evaluations, while helpful in bench-marking each of the techniques independently, cannot alone determine their suitability as part of the attack. We select transforms that show potential against several CMs even if they are not effective against all systems or lead to a slight degradation against some, as the existence of other methods in the full attack can make up for such minor losses. The transformations' order is somewhat deterministic (see §IV-C). Hence, evaluating each transformation's actual potential as part of the attack is better done in a cumulative manner, as is in §VI-A.

We can see that F_1 (silence replacement) is highly effective against almost all CMs. F_2 is not as universal as for three systems (AIR, $MLCG$, $RES2NET$) we observe a mild attack degradation w.r.t the spoofing baseline. Yet, for the majority of systems, F_2 increases the attack's success (often considerably— MCG, $SSNET$, $RAWDARTS$, $AASIST$-L), warranting its inclusion. Similar to F_1, F_3 (spectrum center boosting), F_6 (noise reduction), and F_7 (ADVSR) drastically increase the success rates for several systems and should naturally be included. As for F_4 (local echo), the findings are similar to F_2, being effective especially against CQT-based systems. Finally, F_5 (pre-emphasis) is powerful against $raw\ waveform$-based systems ($SSNET, AASIST$-L, specifically). That said, its potential appears limited compared to when integrated into the full attack (see Table II).

To fathom this, recall that we use a small pre-emphasis coefficient (0.5). As noted in §VI-A and demonstrated in Table XI, F_5 is considerably stronger upon increasing the

Frontend	LPMS	LFCC				CQT				Raw Waveform			
Perturbation \ System	LCNN	AIR	AIR_AM	DARTS	LCNN_LSTM	MCG	MLCG	RES2NET	SSNET	RAWGAT_ST	RAWDARTS	AASIST	AASIST-L
None	3.16%	2.32%	3.59%	4.01%	2.95%	1.90%	4.01%	2.95%	2.74%	**3.16%**	**1.90%**	**2.32%**	2.32%
0.0005	9.28%	7.17%	8.44%	6.12%	3.80%	1.69%	5.49%	2.74%	6.54%	**3.16%**	**1.90%**	**2.32%**	2.32%
0.0015	9.28%	17.09%	14.98%	17.51%	4.85%	4.64%	**9.49%**	**6.54%**	14.98%	**3.16%**	**1.90%**	**2.32%**	**2.53%**
0.0025	9.28%	18.99%	15.61%	19.83%	5.06%	12.24%	6.96%	4.43%	**18.14%**	2.95%	**1.90%**	**2.32%**	**2.53%**
0.0035	**9.49%**	**20.25%**	**15.19%**	**20.46%**	**5.27%**	**18.35%**	6.33%	1.69%	17.09%	2.95%	**1.90%**	**2.32%**	**2.53%**
Best	**9.49%**	**20.25%**	**15.19%**	**20.46%**	**5.27%**	**18.35%**	**9.49%**	**6.54%**	**18.14%**	**3.16%**	**1.90%**	**2.32%**	**2.53%**

Table VIII: CM performance under additive noise. Bold numbers indicate the highest success rate against each system.

Frontend		LPMS	LFCC				CQT				Raw Waveform			
Algorithm	Perturbation \ System	LCNN	AIR	AIR_AM	DARTS	LCNN_LSTM	MCG	MLCG	RES2NET	SSNET	RAWGAT_ST	RAWDARTS	AASIST	AASIST-L
None	None	3.16%	2.32%	3.59%	4.01%	2.95%	1.90%	4.01%	2.95%	2.74%	3.16%	1.90%	2.32%	2.32%
Iterative FGSM	iters=5, eps=0.0005	23.00%	18.14%	15.61%	18.78%	**7.17%**	9.70%	12.03%	**8.86%**	8.86%	**3.16%**	**1.90%**	**2.32%**	2.32%
	iters=5, eps=0.0015	30.80%	30.38%	21.10%	22.15%	6.96%	27.43%	16.24%	2.95%	17.51%	2.95%	**1.90%**	**2.32%**	2.53%
	iters=5, eps=0.0025	41.56%	39.03%	23.84%	22.36%	5.49%	34.39%	23.63%	0.42%	17.30%	2.95%	**1.90%**	**2.32%**	2.53%
	iters=5, eps=0.0035	47.26%	44.51%	26.79%	22.36%	4.22%	41.14%	**26.16%**	0.21%	16.67%	2.74%	**1.90%**	**2.32%**	2.74%
	iters=5, eps=0.0045	49.37%	50.63%	28.69%	23.00%	3.80%	37.13%	24.47%	0.21%	16.67%	2.53%	**1.90%**	1.90%	2.95%
	iters=5, eps=0.0055	51.48%	**54.85%**	**31.86%**	**23.42%**	3.59%	**43.67%**	25.74%	0.00%	16.24%	2.74%	**1.90%**	1.90%	**3.38%**
	iters=10, eps=0.0015	47.68%	35.23%	23.00%	22.36%	7.38%	28.27%	15.40%	2.32%	**17.93%**	2.95%	**1.90%**	**2.32%**	2.53%
	iters=15, eps=0.0015	**55.91%**	36.50%	23.63%	22.15%	**7.17%**	29.32%	14.56%	2.32%	**17.93%**	2.95%	**1.90%**	**2.32%**	2.53%
Carlini & Wagner (learning rate=0.01)	iters=20, confidence=0.0	**22.15%**	**31.01%**	**17.93%**	19.20%	**2.32%**	**25.95%**	**27.85%**	**2.53%**	**11.60%**	1.27%	**1.05%**	**0.84%**	**1.05%**
	iters: 20, confidence=5.0	16.24%	20.89%	17.30%	**19.41%**	1.05%	10.55%	17.51%	1.69%	5.27%	**1.69%**	0.84%	**0.84%**	**1.05%**
PGD (restarts=5)	iters=20, eps=0.003	**70.46%**	**42.62%**	**23.00%**	**21.73%**	**8.23%**	**45.36%**	**16.67%**	1.05%	**15.82%**	2.32%	**1.90%**	**2.32%**	**3.59%**
	iters=20, eps=0.0003	25.53%	12.87%	14.56%	18.14%	6.33%	7.17%	9.07%	**5.06%**	9.28%	**3.16%**	**1.90%**	**2.32%**	2.32%
Best	N/A	**70.46%**	**54.58%**	**31.86%**	**23.42%**	**8.23%**	**43.67%**	**27.85%**	**8.86%**	**17.93%**	**3.16%**	**1.90%**	**2.32%**	**3.59%**

Table IX: CM performance under transferability. Bold numbers indicate the highest success rate against each system.

Frontend	LPMS	LFCC				CQT				Raw Waveform			
Transformation \ System	LCNN	AIR	AIR_AM	DARTS	LCNN_LSTM	MCG	MLCG	RES2NET	SSNET	RAWGAT_ST	RAWDARTS	AASIST	AASIST-L
None	3.16%	2.32%	3.59%	4.01%	2.95%	1.90%	4.01%	2.95%	2.74%	3.16%	1.90%	2.32%	2.32%
F1	**13.71%**	15.61%	13.92%	13.08%	10.97%	12.66%	9.28%	4.85%	12.66%	**7.59%**	1.69%	**4.85%**	**5.27%**
F2	3.16%	2.11%	3.80%	4.22%	3.80%	5.06%	3.38%	1.05%	6.96%	4.85%	2.74%	2.95%	3.16%
F3	4.01%	11.81%	11.81%	6.12%	**21.10%**	6.75%	6.33%	6.33%	2.11%	2.95%	**2.95%**	2.95%	2.32%
F4	2.95%	2.95%	4.01%	4.22%	4.43%	10.97%	6.12%	3.38%	2.53%	3.38%	2.11%	2.53%	2.74%
F5	4.01%	4.64%	4.85%	4.43%	1.90%	3.16%	3.80%	2.11%	5.49%	3.38%	2.11%	3.38%	4.43%
F6	1.48%	1.90%	4.64%	4.22%	3.59%	**15.40%**	6.12%	3.59%	3.59%	2.32%	1.90%	2.95%	4.22%
F7	9.28%	**29.32%**	**27.00%**	**18.57%**	11.18%	8.65%	**16.67%**	**8.65%**	**17.51%**	2.95%	1.90%	2.32%	2.74%

Table X: Independent transformations' success rates against CMs. Bold entries represent the best results for each system.

Frontend	LPMS	LFCC				CQT				Raw Waveform			
System	LCNN	AIR	AIR_AM	DARTS	LCNN_LSTM	MCG	MLCG	RES2NET	SSNET	RAWGAT_ST	RAWDARTS	AASIST	AASIST-L
ASR	0.63%	13.5%	9.07%	0.00%	2.53%	35.65%	9.7%	13.92%	13.5%	10.97%	10.55%	9.49%	12.03%

Table XI: Results of a simple pre-emphasis attack with a large coefficient (0.97).

coefficient to 0.97. In the collective attack, F_5 is coupled with F_3, which itself similarly performs amplification in higher frequencies, serving as a precursor that compensates for using a smaller coefficient, making F_5's effects dominant again. We observe similar results for other transforms that are more powerful when applied jointly, such as F_2 when preceded by F_1 (see Table II). The reasoning is similar, as there is a direct connection between these transformations and while one individually eliminates some machine cues, it requires others' assistance to achieve its full potential.

E. Content Preservation

We use the Python API of Google Cloud STT [36] and the REST API of Microsoft Azure STT [37]. For Google, we use the "video" model for transcription, as it is recommended for inputs with a 16KHz sampling rate, as in our case.

Our attacks assume state-of-the-art spoofing algorithms, such as Shen et al.'s [10] are accessible to the attacker. Such algorithms are known to produce high-quality speech and we assume they can generate speech that is transcribed correctly by STTs. Thus, we randomly select 25 spoofed samples from our high-quality evaluation dataset of 474. For each sample we generate an adversarial using our full attack.

We test whether the STTs transcribe both samples identically. The common metric for evaluating STTs is the word error rate (WER), which is the percentage of words transcribed incorrectly in a given input on average. Since STTs are not perfect and make mistakes often, we cannot assume that if the provided input does not exactly transcribe to the requested text then it will be rejected. Instead, the system must account for its own mistakes. Thus, instead of strictly requiring that the two samples (spoofed and adversarial) transcribe exactly to the same text, we consider the adversarial attack successful if the ratio of different words is below the average WER for each of the STTs. We get the WER values from a benchmark [114].

Results. We achieve an attack success rate of 96% against Microsoft Azure STT and 76% against Google Cloud STT.