On the Detection of Adaptive Adversarial Attacks in Speaker Verification Systems

Zesheng Chen®, Senior Member, IEEE

Abstract—Speaker verification (SV) systems have been widely used in smartphones and Internet of Things devices to identify legitimate users. In recent work, it has been shown that adversarial attacks, such as FAKEBOB, can work effectively against SV systems. The goal of this article is to design a detector that can distinguish an original audio from an audio contaminated by adversarial attacks, if the original audio and the audio before contaminated by the attack have a similar signal-to-noise ratio. Specifically, our designed detector, called MEH-FEST, calculates the minimum energy in an audio signal's high-frequency band through the short-time Fourier transform and uses it as a detection metric. Through both analysis and experiments, we show that our proposed detector is easy to implement, fast to process an input audio, and effective in determining whether an audio is corrupted by FAKEBOB attacks. The experimental results indicate that the detector is extremely effective: with near zero false-positive and false-negative rates for detecting FAKEBOB attacks in Gaussian mixture model (GMM) and i-vector SV systems. Moreover, adaptive adversarial attacks against our proposed detector and their countermeasures are discussed and studied, showing the game between attackers and defenders.

Index Terms—Adaptive attacks, adversarial attacks, detection, energy, short-time Fourier transform (STFT), speaker verification (SV) systems.

I. INTRODUCTION

WITH the popularity of smartphones and Internet of Things (IoT) devices at smart home, voice control has been used in popular virtual assistant technologies, such as Amazon Alexa, Apple Siri, and Google Assistant, and becomes a main interface between humans and devices, because of its convenience and ease of operation. To secure such an interface, speaker verification (SV) systems have been widely applied to verify a user’s identity through their voice before allowing them to access a device. Moreover, some e-banking systems have also authorized visitors through their voices. In other words, human voice has been used as biometrics to distinguish between legitimate users and illegal users.

Two main security attacks have recently emerged to tamper with SV systems. One is called replay attacks that record the legitimate user’s speech and then replay it to fool an SV system [30]. Such a sniffing and spoofing attack requires an attacker to obtain a legitimate user’s audio. The other attack is called adversarial attacks that generate a speech acceptable to the system by adding small and well-designed perturbations to an illegal user’s speech [3], [5]. Such an attack does not need a copy of the legitimate user’s speech and is imperceptible to humans. In this work, we focus on adversarial attacks.

Adversarial attacks were discovered when machine learning classifiers were applied at test time in adversarial setting [2], [7], [19]. With the popularity of deep learning techniques, Goodfellow et al. [10] and Szegedy et al. [26] showed that image classifiers are particularly vulnerable to adversarial attacks. Moreover, it has been found that adversarial attacks can be applied to a wide range of domains such as medical IoT devices [24], image steganography [27], multimedia forensics [4], malware detection [16], and cyber–physical systems [17]. Since the currently widely used SV systems, such as Gaussian mixture model (GMM), i-vector, d-vector, and x-vector, are based on machine learning, they are vulnerable to adversarial attacks. Specially, Chen et al. [5] designed a black-box adversarial attack, called FAKEBOB, that does not require the implementation information of an SV system and only needs the output scores of the system. FAKEBOB is shown to be very effective against both open-source and commercial systems, and can achieve 99% targeted attack success rate (ASR). Moreover, it has been shown in [5] that several defense methods, including local smoothing, quantization, and temporal dependency detection [32], that work well against adversarial attacks in the image domain are not able to counteract FAKEBOB.

There have been some recent works proposed to detect or defend against adversarial attacks in SV systems [1], [3], [8], [12], [13], [18], [29], [31]. For example, Li et al. [18] used a separate neural network to detect adversarial samples. Joshi et al. [13] studied generative adversarial networks (GANs) based and variational autoencoders (VAEs) based defenses. Wu et al. [31] proposed to sample the neighbors of a given speech and calculate the average score based on these neighbors. In our previous work, we designed a defense system by adding small random Gaussian noise to an input audio [3]. However, the following important research question still remains: How can we effectively and efficiently distinguish between an original audio and an adversarial audio? The original audio is either from a legitimate user or from an illegal user and is without perturbations, whereas the adversarial audio can be either a successful attack or a failed attack and is with the attacker’s designed perturbations.
Our previous work [3] has shown that the perturbations introduced by the adversarial attack are similar to white noise. This result motivated us with the following question: if an audio from an illegal user, before contaminated by an adversarial attack, has a similar signal-to-noise ratio (SNR) as the audio from a legitimate user, how can we accurately detect the adversarial audio (i.e., the illegal audio after contaminated by the adversarial attack) in real time? This question leads to our work that aims to design a simple, fast, and effective detector to determine whether a given audio has been processed by an adversarial attack such as FAKEBOB. To achieve this goal, we propose a new detector called Minimum Energy in High Frequencies for Short Time (MEH-FEST). We give such a name in hope to bring disappointment (i.e., MEH) to attackers and to focus on the most important feature of an audio signal (i.e., FEST). Our designed MEH-FEST detector is based on the following three key observations.

1) Although small, perturbations in adversarial attacks against SV systems behave like white noise and are present everywhere in the audio signal across time and frequency.

2) Audio signals are nonstationary [20]. If audio signals from a speaker are divided into small pieces based on a short-time period (e.g., 32 ms), these short-time periods always contain cases when speech is absent or very weak. As a result, the way that perturbations affect audio signals is significantly different when speech is present or absent.

3) The energy in the high-frequency range of an original audio is usually small, especially when speech is absent or very weak.

Specifically, the MEH-FEST detector applies short-time Fourier transform (STFT) [21] and calculates the minimum energy of an audio signal in the high-frequency range among short-time periods. Through analysis and experiments, we demonstrate that our proposed MEH-FEST detector is as follows.

1) Simple: The method simply focuses on the key information of a given audio and is easy to implement.

2) Fast: The detector can process an input audio extremely fast, e.g., within 3.4 ms in our experiments.

3) Effective: As shown in the experiments, the MEH-FEST detector is with a false-positive rate (FPR) of 0% and false-negative rates (FNRs) of 0.053% and 0% for detecting FAKEBOB attacks in GMM and i-vector SV systems, respectively.

For a theoretical analysis, we estimate the energy of an audio with the attacker’s perturbations in the high-frequency range for a short-time period when speech is absent. In particular, we show analytically that the MEH-FEST metric is able to amplify the variance of perturbations so that the perturbations are more detectable. Moreover, our analysis establishes a relationship between the standard deviation of perturbations and the detection threshold of MEH-FEST, and identifies the theoretical criteria for the effective operation of our proposed method. The experimental results verify our theoretical analysis.

Inspired by the adaptive attacks proposed in [28], we further consider how attackers can design a white-box countermeasure to avoid the detection of the MEH-FEST method. Specifically, two different types of adaptive attacks were studied. One attempts to reduce the perturbation threshold of an attack. However, through experiments, we found that such an adaptive attack would reduce the attacking power and is meanwhile vulnerable to the noise-adding defense method proposed in our previous work [3]. The other adaptive attack attempts to avoid perturbing the short-time period signal in an audio and, thus, keeps the same minimum energy calculated by the MEH-FEST method. For this attack, we then propose a countermeasure that measures the second minimum energy, which can also effectively distinguish between an original audio and an adversarial audio, in the similar way as using the minimum energy. Such a game between attackers and defenders continues when both sides can obtain the implementation details of the other side. In our experiments, we demonstrate the performance of adaptive FAKEBOB attacks and the corresponding countermeasures against them.

It is noted that the interaction between attackers and defenders can be viewed from the perspective of game theory, especially repeated games [43]. On the one hand, defenders attempt to understand the methods that attackers apply and develop systems against them. On the other hand, attackers endeavor to learn the weakness of the defense systems and design new ways to evade these systems. Repeatedly, both attackers and defenders can learn about the opponent’s strategies and, thus, design the corresponding countermeasures.

The main contributions of our work are summarized as follows.

1) We design a new detector, i.e., MEH-FEST, to distinguish between original audios and adversarial audios. Through both analysis and experiments, we show that if legitimate audios and illegal audios (i.e., audios before contaminated by adversarial attacks) have a similar SNR, our designed detector can effectively identify the adversarial audios.

2) We study and discuss adaptive adversarial attacks against our proposed MEH-FEST detection method and their countermeasure, and demonstrate a game between attackers and defenders.

The remainder of this article is structured as follows. Section II reviews related background and presents important observations that lead to our design. Section III provides the implementation details of the MEH-FEST detector, whereas Section IV gives the theoretical analysis of our proposed method. Next, Section V discusses two possible adaptive adversarial attacks against our MEH-FEST method and our proposed countermeasures against these adaptive attacks. Section VI evaluates through experiments the performance of the MEH-FEST method against FAKEBOB attacks, as well as our countermeasures against adaptive FAKEBOB attacks. Finally, Section VII concludes this article and discusses the future work.

II. BACKGROUND AND OBSERVATIONS

A. Audio Signal

It is well known that the hearing frequency range of an audio signal for humans is roughly between 20 Hz and 20 kHz [38].
Moreover, the audio signal changes with time. Fig. 1(a) plots a waveform of an example audio signal, showing how the amplitude of the audio signal varies with time.

One key observation of audio signals is that they are non-stationary [20]. That is, the statistical properties of the audio signal change with time. In particular, an audio signal is very different when speech is present or absent. For example, Fig. 1(b) and (c) shows the same audio signal in Fig. 1(a) in two different short-time periods: from 3.5 to 3.532 s and from 3.98 to 4.012 s, respectively. There is a notable difference in amplitude values between these two time periods. The amplitude in Fig. 1(b) is between –0.27 and 0.22, reflecting the presence of speech; but the absolute value of amplitude in Fig. 1(c) is very small and less than 0.0008, indicating the absence of speech.

In this work, we focus on a digital audio signal, which is stored in computers in digital format and is mathematically denoted by \( s[n] \), \( n = 0, 1, 2, \ldots \). Assuming that the sampling frequency is \( f_s \), the relationship between the discrete index \( n \) and the continuous time index \( t \) is \( t = n/f_s \). For example, when \( f_s = 16 \) kHz and \( t = 3.5 \) s, \( n = t \times f_s = 56,000 \).

B. FAKEBOB Attacks Against Speaker Verification Systems

An SV system has been applied to determine whether a user is legitimate or illegal. Currently, the most widely used SV systems, such as GMM [25] and i-vector [9], are score based. Specifically, the score-based SV system provides a function, \( S \), that calculates the score of a given input audio \( s[n] \) and then compares the score with a threshold, \( \theta \). If \( S(s[n]) \geq \theta \), the SV system would accept \( s[n] \); otherwise, it would reject \( s[n] \). There are two main performance metrics for an SV system. One is the false acceptance rate (FAR), which indicates the percentage of audios from an illegal user that are falsely accepted by the system. The other is the false rejection rate (FRR), which presents the percentage of audios from a legitimate user that are falsely rejected by the system. The threshold of the SV system, i.e., \( \theta \), is determined when FAR is equal to FRR, which is called the equal error rate (EER) [6]. A smaller EER reflects a better SV system.

An attacker can design an adversarial example attack to make the SV system falsely accept an illegal user as a legitimate user. FAKEBOB is the state-of-the-art black-box adversarial attack against popular score-based SV systems such as GMM and i-vector [5]. The basic idea of FAKEBOB attacks is to find small perturbations \( p[n] \), so that an SV system would reject \( s[n] \), but accept \( a[n] \), where \( s[n] \) is the audio from an illegal user and \( a[n] = s[n] + p[n] \). Here, we call \( s[n] \) as the original illegal audio and \( a[n] \) as the adversarial audio. To make the audio imperceptible to humans (i.e., adversarial audio \( a[n] \) sounds like illegal audio \( s[n] \)), it requires that \( |p[n]| \leq \epsilon \), where \( \epsilon \) is called the perturbation threshold and should be small. Specifically, FAKEBOB applies the basic iterative method (BIM) [14] and the natural evolution strategy (NES) [11] to find the optimal \( p[n] \). In other words, FAKEBOB attempts to estimate the gradient decent of the objective function over the input audio to find the direction to change the audio and apply multiple iterations to create an adversarial audio. The objective function for FAKEBOB attacks is

\[
L(a[n]) = \max[\theta - S(a[n]), 0]
\]  

and the gradient decent function over the input audio is

\[
f_G(a[n]) = \nabla a[n]L(a[n]).
\]

Moreover, FAKEBOB uses a sign function

\[
f_S(x) = \begin{cases} 
1, & \text{if } x > 0 \\
0, & \text{if } x = 0 \\
-1, & \text{if } x < 0 
\end{cases}
\]

and a clip function

\[
f_C(a[n]) = \begin{cases} 
a[n], & \text{if } |a[n] - s[n]| < \epsilon \\
|s[n] + \epsilon|, & \text{if } a[n] \geq s[n] + \epsilon \\
|s[n] - \epsilon|, & \text{if } a[n] \leq s[n] - \epsilon 
\end{cases}
\]

Applying these three functions, FAKEBOB updates the input adversarial audio \( a[n] \) through the following operation:

\[
a[n] \leftarrow f_C(a[n] - lr \times f_S(f_G(a[n])))
\]

where \( lr \) is the learning rate and can change based on the status of iterations. The implementation of the FAKEBOB attack is summarized in Algorithm 1. In [5], FAKEBOB is shown to be able to achieve a very high targeted ASR on popular SV systems.

Using the audio in Fig. 1(a) as the original illegal audio, FAKEBOB is able to generate a successful adversarial audio against a GMM SV with \( \epsilon = 0.002 \), shown in Fig. 2(a). The waveforms in Figs. 1(a) and 2(a) are very similar with indistinguishable differences to humans. However, as a result of the
Comparing the nonstationary property of an audio, the effect of perturbations is significantly different when speech is present or absent. We show the waveform of this adversarial audio in two distinct short-time periods, i.e., from 3.5 to 3.532 s and from 3.98 to 4.012 s when speech is absent. On the other hand, from Figs. 1(c) and 2(c), it is clear that when speech is absent, the impact of the perturbations is very minor. Based on this observation, our designed detector attempts to distinguish between the original audio and the adversarial audio based on the time period when speech is absent.

C. Short-Time Fourier Transform

STFT is a widely used tool for studying audio signals [20], [21]. Specifically, an audio signal \( s[n] \) can be transformed into the frequency domain by the following equation:

\[
S[k, m] = \sum_{n=0}^{N-1} w[n] s[n + mH] e^{-j2\pi kn/N}, \quad k = 0, 1, \ldots, N/2
\]

where \( w[n] \) is called the analysis window (e.g., Hann window) and is used to avoid the ripple artifacts. The analysis window is with a length of \( W \), during which the statistical property of the audio signal does not change much. \( N \) is the length for the fast Fourier transform (FFT) and is assumed to be a power of two [20]. Note that \( N \geq W \). \( H \) is called the hop size and is used to specify the step size in which the window is to be shifted across the signal [21]. \( m \) is a nonnegative integer and takes values from 0 to \( \lfloor (L - N)/H \rfloor \), where \( L \) is the length of the digital audio signal.

As a result of STFT, \( S[k, m] \) contains both time and frequency information, as \( k \) refers to the frequency and \( m \) refers to the time. A mel spectrogram has been widely used to virtualize the magnitude of the spectrum, i.e., \( |S[k, m]| \) [20], [21]. As shown in Fig. 3, the \( x \)-axis of the mel spectrogram is the time, the \( y \)-axis is the frequency in a log scale, and the color represents the magnitude in dB. We plot the mel spectrograms of the original audio [i.e., \( s[n] \) in Fig. 1(a)] and the adversarial audio [i.e., \( a[n] \) in Fig. 2(a)] in Fig. 3(a) and (b), respectively. The audios are with \( f_s = 16 \) kHz. The STFT in these mel spectrograms uses a Hann window with size \( W = 400 \) that is equal to a duration of 25 ms, the FFT length \( N = 512 \) corresponding to 32 ms, and the hop size \( H = 160 \) that is equivalent to 10 ms. It can be seen that although these two mel spectrograms are similar, the background blue color for the adversarial audio is lighter than that for the original audio, indicating more energy in the...
background for the adversarial audio. We further plot the mel spectrogram of the perturbations (i.e., $p[n]$) in Fig. 3(c). It is evident that the blue color spreads evenly across time and frequency, indicating that the perturbations behave in a similar way as white noise.

Furthermore, the mel spectrogram of the original audio in Fig. 3(a) indicates that in general, the magnitude of the signal at the higher frequency range is much smaller than that of lower frequency content. Moreover, comparing the original audio with the adversarial audio in the high-frequency range, we observe that the magnitude of the adversarial audio is obviously larger than that of the original audio. This observation inspires us to focus our detector on the high-frequency range.

Fig. 4. Magnitude of the STFT of the short-time original and adversarial audios, and energy in the high-frequency range over time frames. (a) Original audio (when speech is absent). (b) Adversarial audio (when speech is absent). (c) Energy in high frequencies over time frames.

III. MEH-FEST DETECTOR

The MEH-FEST detector attempts to perform a hypothesis test to decide whether an audio is an original audio (either from a legitimate user or from an illegal user) or an adversarial audio (either a successful attack or a failed attack), as shown in the following hypothesis:

$$
H_0 : \text{the audio is an original audio} \\
H_1 : \text{the audio is an adversarial audio}.
$$

The main performance metrics to evaluate a detector include the false positive rate $P_{FP} = P(H_1|H_0)$ and the false negative rate $P_{FN} = P(H_0|H_1)$. The goal of our designed detector is to make both $P_{FP}$ and $P_{FN}$ as small as possible. Two other performance metrics are the true positive rate $P_{TP} = P(H_1|H_1)$ and the true negative rate $P_{TN} = P(H_0|H_0)$. It is noted that $P_{TP} = 1 - P_{FP}$ and $P_{TN} = 1 - P_{FN}$.

To further understand the effect of perturbations on signals in absence of speech, we plot the magnitude of the STFT of both the short-time original audio [from Fig. 1(c)] and the short-time adversarial audio [from Fig. 2(c)] in Fig. 4(a) and (b), respectively. Since these two short-time audios last only 32 ms that is equal to the time length for $N$, $m$ is a fixed number in $S[k, m]$, and we can plot how the magnitude varies with the frequency in Fig. 4(a) and (b). It can be seen that for the original audio, the magnitude is very small when the frequency is high. On the other hand, the magnitude of the adversarial audio is much larger than that of the original audio at the high-frequency range. Based on this observation, we calculate the energy of the audio signal among the high-frequency range, i.e.,

$$
E_r[m] = \sum_{k \geq f_t} |S[k, m]|^2
$$

where $f_t$ is the frequency threshold to determine the high-frequency range. For example, if we consider the high-frequency range above 7 kHz, $f_t = (N/2) \times (7k/8k) = 224$ and, thus, the energy of the original audio in Fig. 4(a) can be calculated as $E_r[m] = 7.7 \times 10^{-6}$, whereas for the adversarial audio in Fig. 4(b), $E_r[m] = 5.6 \times 10^{-3}$. Through this example we illustrate that the energy in the high-frequency range is significantly different for these two audios when speech is absent.

How can we find the time frame (i.e., $m$) in which the speech is absent? To identify a proper $m$, we plot how $E_r[m]$ varies with $m$ for both original and adversarial audios [from Figs. 1(a) and 2(a)] in Fig. 4(c). In this figure, the $y$-axis uses a log scale to enhance the visibility of the differences between these two audios. It can be seen that in many time frames, $E_r[m]$ of the adversarial audio is larger than that of the original audio. Most importantly, in all time frames, $E_r[m]$ of the adversarial audio is no less than $1.9 \times 10^{-3}$, whereas the minimum of $E_r[m]$ for the original audio is only $7.7 \times 10^{-6}$. This provides a heuristic that when $E_r[m]$ is minimal, the corresponding $m$ indicates the time frame in which the speech is absent. Therefore, we have found a metric that can be used for our MEH-FEST detector

$$
E = \min_m E_r[m] = \min_m \sum_{k \geq f_t} |S[k, m]|^2.
$$

Essentially, our detector calculates the minimum energy $E$ in the high-frequency range for the STFT of an audio and uses it to make a decision as follows:

\[
\begin{aligned}
\text{If } E \leq D, & \text{ the audio is an original audio } (H_0) \\
\text{If } E > D, & \text{ the audio is an adversarial audio } (H_1)
\end{aligned}
\]

where $D$ is called the detector threshold and is a user selected constant. A proper value of $D$ is important for the MEH-FEST detector. If $D$ is too large, the detector would miss detecting many adversarial audios and, thus, lead to a large FNR. Otherwise, if $D$ is too small, our MEH-FEST detector would treat many original audios as malicious and, thus, cause a large FPR.

How can we find a proper value for $D$? In the perspective of machine learning, we can regard our detection problem as an
unsupervised machine learning problem and estimate $D$ from the existing, trusted original audios, either from legitimate users or from illegal users. Specifically, before applying the MEH-FEST detector to test an audio, we calculate $E$ in (8) for a list of trusted original audios. We then find the mean value and the standard deviation of these $E$’s, which are denoted as $u_E$ and $\sigma_E$, respectively. Thus, $D$ can be estimated by the following:

$$D = u_E + k\sigma_E$$

where $k$ is a controllable parameter to adjust the detection threshold. The selection of $k$ affects $P_{FP}$ and $P_{FN}$. If $k$ is too small (e.g., $k = 0$), it would lead to a large $P_{FP}$. On the other hand, if $k$ is too large (e.g., $k = 5$), it may potentially result in a large $P_{FN}$. In this work, we choose $k = 3$, as inspired by the three sigma limits statistical calculation [36]. For example, if $E$ follows a normal distribution for original audios, the FPR is only 0.3% [42].

In summary, the MEH-FEST detection method is given in Algorithm 2.

### Algorithm 2 MEH-FEST Detection Method

1. **Input:** input audio $s[n]$, analysis window $w[n]$, window length $W$, FFT length $N$, hop size $H$, frequency threshold $J$, and detection threshold $D$

2. **Output:** whether the input audio $s[n]$ is an adversarial audio

3: Calculate the STFT $S[k, m]$ of input audio $s[n]$ based on $w[n]$, $W$, $N$, and $H$ from Equation (6)

4: $E = \min_{m} \sum_{k \geq l_{2}} |S[k, m]|^2$

5: if $E > D$ then

7: return true

8: else

9: return false

end if

In this section, we provide a theoretical analysis of our designed MEH-FEST detector. Specifically, we quantitatively analyze the effect of perturbations on the energy $E$ in (8) when speech is absent and there is no or ignorable background noise. We consider two cases: 1) single short-time frame and 2) multiple short-time frames. Next, we extend our study to the case there is unignorable background noise in original audios.

#### A. Single Short-Time Frame

Based on the observation from Fig. 3(c), we assume that adversarial perturbations $p[n]$’s are independent and identically distributed (i.i.d.) random variables that follow a normal distribution with zero mean and $\sigma^2$ variance, i.e.,

$$p[n] \sim N(0, \sigma^2).$$

That is, $E[p[n]] = 0$, and $E[p^2[n]] = \sigma^2$. Moreover $E[p[n]p[l]] = 0$, when $n \neq l$.

Here, the standard deviation $\sigma$ is affected by the perturbation threshold $\epsilon$. When $\epsilon$ increases, $\sigma$ also increases. Furthermore, since $|p[n]| \leq \epsilon$, $\sigma^2 = E[p^2[n]] \leq \epsilon^2$, which means $\sigma \leq \epsilon$.

We study the short-time period when speech is absent in this section. In the absence of speech, the original audio signal is very small and is assumed to be zero, i.e., $s[n] = 0$. As a result, the adversarial audio $q[n] = s[n] + p[n] = p[n]$, containing only perturbations.

Applying the STFT in (6) to an adversarial audio for a single short-time frame when $s[n] = 0$, we have

$$S[k] = \sum_{n=0}^{N-1} w[n]p[n]e^{-j2\pi kn/N}, \quad k = 0, 1, \ldots, N/2.$$  \hspace{1cm} (13)

Note that the expectation of $S[k]$ is zero. Setting $q[n] = w[n]p[n]$, we derive the expectation of $|S[k]|^2$ in the following:

$$E\left[|S[k]|^2\right]$$

$$= E\left[\left(\sum_{n=0}^{N-1} q[n]e^{-j2\pi kn/N}\right)^2\right]$$

$$= \sum_{n=0}^{N-1} E[q[n]^2]e^{-j2\pi k(n-\ell)/N}. \hspace{1cm} (14)$$

Since $q[l]$ is a real signal, $\overline{q[l]} = q[l]$. Moreover, when $n \neq \ell$, $E[q[n]q[l]] = w[n]w[\ell]E[p[n]p[l]] = 0$ based on (12). Therefore

$$E\left[|S[k]|^2\right] = \sum_{n=0}^{N-1} E\left[q^2[n]\right] \hspace{1cm} (15)$$

$$= \sigma^2 \sum_{n=0}^{N-1} w^2[n]. \hspace{1cm} (16)$$

Analysis window $w[n]$ can take different forms. In this work, we apply the widely used Hann window [35] as the analysis window, i.e.,

$$w[n] = 0.5 - 0.5 \cos\left(\frac{2\pi n}{W-1}\right) , \quad 0 \leq n \leq W-1. \hspace{1cm} (17)$$

Using the continuous-time integral as an approximation to the discrete-time summation, we find that

$$\sum_{n=0}^{N-1} w^2[n]$$

$$\approx \int_{0}^{W-1} 0.5 - 0.5 \cos\left(\frac{2\pi x}{W-1}\right)^2 dx \hspace{1cm} (18)$$

$$= \frac{W-1}{8\pi} \int_{0}^{2\pi} [1 - \cos(y)]^2 dy \hspace{1cm} (19)$$

$$= \frac{3}{8}(W-1). \hspace{1cm} (20)$$

Therefore

$$E\left[|S[k]|^2\right] = \frac{3}{8}(W-1)\sigma^2. \hspace{1cm} (21)$$

Putting the above equation into (8), we have

$$E = \sum_{k \geq l_{2}} E\left[|S[k]|^2\right] = \frac{3}{8}(W-1)M\sigma^2 \hspace{1cm} (22)$$

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
where \( M = N/2 - f_i + 1 \), indicating the number of discrete frequency content in high frequencies above \( f_i \).

Equation (22) provides the expected value of the energy in the high-frequency range for a single short-time frame when speech is absent. In an audio, however, there are multiple short-time frames when speech is absent or the energy of speech in the high-frequency range is minimal. Since \( E \) is the minimum value of the multiple energies in these time frames as shown in (8), \( E \leq 3(W - 1)M\sigma^2/8 \).

### B. Multiple Short-Time Frames

We consider multiple short-time frames when speech is absent or the energy of the speech in the high-frequency range is very small, i.e., \( m = 0, 1, 2, \ldots, F - 1 \), where \( F \) is the number of short-time frames considered and \( F > 1 \). Since in (6) \( s[n + mF] = p[n + mF] \) when speech is absent or the energy of the speech among the high-frequency range is very small, \( s[n + mF]'s \) are i.i.d. random variables following a normal distribution with zero mean and \( \sigma^2 \) variance. Moreover, in STFT \( S[k, m]'s \) are a linear combination of \( s[n + mF]'s \). As a result, \( S[k, m]'s \) are random variables that follow a complex normal distribution [40] with zero mean and a covariance of \( \Gamma = 3(W - 1)\sigma^2/8 \) based on (21). Furthermore, \( E_r[m] \) in (7) is the sum of the squares of \( M \) normal random variables. If we set

\[
X_r[m] = \frac{8E_r[m]}{3(W - 1)\sigma^2} = E_r[m]/\Gamma
\]  

(23)

then \( X_r[m]'s \) follow the chi-squared distribution [39] with \( M \) degrees of freedom, i.e.,

\[
X_r[m] \sim \chi^2_M.
\]  

(24)

Note that

\[
E = E\left[ \min_m E_r[m] \right] = \Gamma E\left[ \min_m X_r[m] \right]
\]  

(25)

where \( m = 0, 1, 2, \ldots, F - 1 \). When \( F = 1 \), \( E[\min_m X_r[m]] = E[X_r[m]] = M \), so \( E \) in (25) is reduced to that in (22).

To find the expected value of our MEH-FEST metric for an audio theoretically, we can start with a list of chi-squared distributions [39] with \( M \) degrees of freedom, i.e., \( E[\min_m X_r[m]] \). As a result, \( \Gamma E[\min_m X_r[m]] \) is the theoretical value of the MEH-FEST metric.

Considering the detector threshold \( D \) from (10), if we set \( E = D \) in (25), we then find

\[
\sigma_D = \sqrt{\frac{8D}{3(W - 1)E[\min_m X_r[m]]}}
\]  

(26)

where \( \sigma_D \) is the value of \( \sigma \) that can lead to an energy level of \( D \) in the high-frequency range when speech is absent. As a result, when an adversarial audio has perturbations that are with standard deviation \( \sigma \) larger than \( \sigma_D \), our MEH-FEST method can detect it correctly with a high probability. On the other hand, when \( \sigma < \sigma_D \), the adversarial audio may be able to confuse the MEH-FEST detector, which will be further discussed in the next section.

### C. Unignorable Background Noise

In the previous analysis, we assume that there is no or ignorable background noise in an original audio. Here, we extend our study to consider background noise that is inacceivable. We assume that the background noise is white with \( \sigma_B^2 \) variance. As a result, for an original audio, the MEH-FEST metric \( E \) can be obtained through (27) by replacing \( \sigma^2 \) with \( \sigma_B^2 \). For an adversarial audio, when speech is absent, the original signal \( s[n] \) is a background noise signal and, thus, the variance of the adversarial signal \( a[n] \), i.e., \( \sigma_A^2 \), becomes

\[
\sigma_A^2 = \sigma_B^2 + \sigma^2
\]  

(28)
where $\sigma^2$ is the variance of the perturbation signal $p[n]$. Similarly, the MEH-FEST metric $E$ of an adversarial audio can be derived through (27) by replacing $\sigma^2$ with $\sigma_A^2$.

We would expect that the average of $E$ for adversarial audios is larger than that for original audios. On the one hand, when $\sigma_B$ is much larger than $\sigma$, $\sigma_A$ is close to $\sigma_B$, so that our MEH-FEST method may not be able to distinguish between original audios and adversarial audios. On the other hand, when $\sigma_B$ is much smaller than $\sigma$, $\sigma_A$ is dominated by $\sigma$, and as a result, our proposed MEH-FEST algorithm can perform well.

V. ATTACKERS’ COUNTERMEASURES AND DEFENDERS’ COUNTERMEASURES

If the implementation detail of the MEH-FEST detector is known to an attacker, how can this attacker design a white-box countermeasure to avoid the detection? Such a countermeasure is called an adaptive attack [28]. In this section, we study two possible adaptive attacks, as well as approaches by defenders against these countermeasures.

A. Reducing the Perturbation Threshold $\epsilon$

As shown in Section IV, one intuitive countermeasure by attackers is to reduce the perturbation threshold $\epsilon$ so that the standard deviation of perturbations (i.e., $\sigma$) can be less than $\sigma_0$ in (26). However, as shown in [5] and in our experiments in Section VI-C, when $\epsilon$ decreases, the ASR of an adversarial attack would decrease as well. That is, a small value of $\epsilon$ can let adversarial attacks reduce the attacking power in the first place. Moreover, when $\epsilon$ is small, the resulting adversarial audios may be vulnerable to countermeasures such as the noise-adding defense system proposed in our previous work [3]. As a result, it is not desirable for attackers to use a very small value for $\epsilon$. This will be verified by the experimental results in Section VI-D1.

B. nth FAKEBOB Attacks

If a FAKEBOB attacker knows that our MEH-FEST detector looks for the short-time period that leads to the minimum energy $E$ in (8), the attacker can avoid to perturb that specific short-time period during the iteration process. Specifically, before applying the FAKEBOB attack, an attacker uses a method similar to the MEH-FEST method in Algorithm 2 to the original illegal audio, in order to identify the time frame $T$ that leads to $E$, i.e.,

$$T = \arg \min_m E_r[m] = \arg \min_m \sum_{k \geq f_t} |S[k, m]|^2.$$  

Then, in the execution of the FAKEBOB attack, it avoids to perturb the short-time period between $T \cdot H$ and $T \cdot H + N - 1$ in original audio $s[n]$, where $H$ is the hop size and $N$ is the FFT length in (6). As a result, it is expected that the resulting adversarial audio $a[n]$ would have the same value of $E$ as that of the original audio $s[n]$. The attacker’s countermeasure is summarized in Algorithm 3, which we also call as the 1st FAKEBOB attack.

How can a defender counteract such an adaptive attack? Note that $E_r[m]$ in (7) covers a list of the energy of the audio signal among the high-frequency range over time frames $m$, and $E$ is the smallest element in $E_r[m]$. When an adaptive attack applies Algorithm 3, $E$ would be the same for both original illegal audio $s[n]$ and adversarial audio $a[n]$. To defend against such an attack, an idea is to consider the second minimum element in $E_r[m]$, i.e.,

$$E_2 = \min_m \{e : e \in E_r[m] \setminus \{E \}, e > E \}.$$  

Here, we assume that elements in $E_r[m]$ are distinct. From Fig. 4(c), it can be seen that for many time frames, the value of $E_r[m]$ is near the minimum value for either the original or the adversarial audios. As a result, we can expect that $E_2$ can be used to distinguish between original audios and adversarial audios. The countermeasure against the adaptive attack in Algorithm 3 is summarized in Algorithm 4, which we also refer to as the 2nd MEH-FEST detection method. Note that the detection threshold $D_2$ is different from $D$, but it can be
calculated in a similar way as $D$ by applying (10), where $u_E$ and $\sigma_E$ are obtained based on $E_2$ values, instead of $E$ values, from a list of training original audios.

The game between attackers and defenders can continue. If the implementation of both MEH-FEST in Algorithm 2 and its extension in Algorithm 4 is known to attackers, they would avoid perturbing two short-time frames that lead to $E$ and $E_2$.

Specifically, we define

$$T_2 = \arg \min_m \{e : e \in E_r[m], e > E\}. \quad (31)$$

Then, the attackers generate FAKEBOB adversarial audios by avoiding changing the audio signal in both $[T \cdot H, T \cdot H + N)$ and $[T_2 \cdot H, T_2 \cdot H + N)$. In such a way, both $E$ and $E_2$ in an adversarial audio would be the same as the original audio.

As a countermeasure by defenders, they would look into the third smallest element in $E_r[m]$, i.e., $E_3$, as the detection target. However, attackers can design an adaptive attack that avoid all three time frames that lead to $E$, $E_2$, and $E_3$. We name such a detection method by defenders as $n$th MEH-FEST, where $n$ refers to the $n$th smallest element in $E_r[m]$ and the detection target is $E_n$, i.e.,

$$E_n = \min_m \{e : e \in E_r[m], e > E_{n-1}\}, \quad n = 2, 3, 4, \ldots \quad (32)$$

and $E_1 = E$. Similarly, we name the corresponding adaptive attack as $n$th FAKEBOB, where

$$T_n = \arg \min_m \{e : e \in E_r[m], e > E_{n-1}\}, \quad n = 2, 3, 4, \ldots \quad (33)$$

and $T_1 = T$. It is noted that 1st MEH-FEST is the original MEH-FEST method proposed in Algorithm 2, whereas the 1st FAKEBOB is the FAKEBOB attack with the countermeasure against the 1st MEH-FEST shown in Algorithm 3.

We expect that as $n$ increases, in general the $n$th FAKEBOB attack would reduce the ASR and increase the running time to generate adversarial audios, because more short-time frames are unchanged from the original audios. On the other hand, we also expect that the efficiency of the $n$th MEH-FEST detection method would reduce as $n$ increases, because the values of $E_n$ for original and adversarial audios get closer as $n$ grows.

Note that when calculating the STFT of an audio, i.e., $S[k, m]$, there is signal overlapping between two neighboring time frames when $H < N$. That is, $S[k, m]$ and $S[k, m + 1]$ are calculated with some common $S[n]$’s. Therefore, when $T$ and $T_2$ are neighboring time frames, the efficiency of applying $E_2$ in (30) through Algorithm 4 would be negatively affected against the 1st FAKEBOB attack. To avoid such an effect, we, as a defender, introduce a constraint that

$$|T_2 - T| \geq \lfloor N/H \rfloor \quad (34)$$

in Algorithm 4. That is, we would keep searching for the second smallest element in $E_r[m]$ only for those time frames that are at least $\lfloor N/H \rfloor$ distance from $T$. Moreover, similar constraints can be applied to the $n$th MEH-FEST. For example, for the 3rd MEH-FEST, besides Inequality (34), the following constraints should also be followed: $|T_3 - T| \geq \lfloor N/H \rfloor$, and $|T_3 - T_2| \geq \lfloor N/H \rfloor$.

VI. PERFORMANCE EVALUATIONS

In this section, we first describe the experimental setup. We then verify the analytical results of our designed MEH-FEST method through experiments. Next, we evaluate the performance of the MEH-FEST detector against FAKEBOB attacks. Finally, we show the performance of defenders’ countermeasures against attackers’ adaptive attacks.

A. Experimental Setup

We used a virtual machine (VM) in Google Cloud Platform [34] to run all our experiments. The VM is with 16 cores, 64-GB memory, and 3.10-GHz CPU (i.e., c2-standard-16 machine type) and is installed with Ubuntu 20.04. Moreover, we applied the code and the data set provided in [5] to run FAKEBOB attacks against both GMM and i-vector SV systems, which were implemented by the Kaldi speech recognition toolkit [23]. Specifically, the data set comes from LibriSpeech [22] and contains the audios of five legitimate users and four illegal users. All audios are with a sampling frequency of 16 kHz. There are 25 audios for each illegal user and a total of 500 audios from all legitimate users. It is noted that among these 600 original audios, some audios contain perceptible background noise.

In our MEH-FEST detector shown in Algorithm 2, we chose the following parameters for the STFT: Hann window with a window length $W = 400$ (i.e., 25 ms), FFT length $N = 512$ (i.e., 32 ms), and hop size $H = 160$ (i.e., 10 ms). These parameters have been widely applied to calculate the STFT of an audio signal [15], [21]. We applied the Librosa library [37] to implement the STFT. Moreover, we used 7 kHz as the frequency threshold, i.e., $f_t = 224$, so that the high-frequency range is between 7 and 8 kHz.

B. Verification of Analytical Results

To verify the analytical results of the MEH-FEST detector provided in Section IV, we added Gaussian noise with zero mean and $\sigma^2$ variance upon 600 original audios. We calculated the STFT $S[k, m]$ of an audio with noise based on (6) through the Librosa library and then measured $E$ based on (8). Fig. 6 shows how the average of the measured $E$’s over 600 audios varies with $\sigma^2$, when $\sigma$ increases from 0 to 0.005.
It is noted that the average length of these 600 audios is 4.95 s, which correspond to an average number of samples of 79219.61, i.e., $L = 79219.61$. Therefore, we can estimate $F = [(L - N)/H] + 1 = 492$. As shown in (27) and Fig. 5, the corresponding theoretical value of $E/\sigma^2$ when $F = 492$ is 2057. It can be seen from Fig. 6 that the theoretical $E$ accurately predicts the value of the measured $E$.

### C. Performance of the MEH-FEST Detector Against FAKEBOB Attacks

In the FAKEBOB attack, we used 1000 for the maximum iteration (i.e., $I$) as suggested in [5]. Different from the experiments in [5], we registered each legitimate user in a stand-alone SV system and obtained five different SV systems, instead of registering all five legitimate users into the same SV system. As a result, the performance of FAKEBOB attacks is different from that presented in [5]. However, we consider that such a setup is more realistic. It is noted that the proposed detector can achieve 100% detection rate or true positive rate with our proposed MEH-FEST detector, i.e., $P_{FP} = 0$. Since $P_{TN} = 1 - P_{FP}$, the true-negative rate is 1.

Next, we consider the FNR and plot the CDF of $E$ for FAKEBOB adversarial audios against GMM SV systems with different $\epsilon$, i.e., $\epsilon = 0.0005, 0.001, 0.002$, and 0.005, in Fig. 7(b). Note that in this figure, the $x$-axis uses a log scale. It can be seen that in general, when $\epsilon$ increases, the CDF of $E$ shifts to the right, indicating an overall increase of $E$. Moreover, when $\epsilon \geq 0.001$, all values of $\epsilon$ are greater than $D$ (i.e., $1.52 \times 10^{-4}$). When $\epsilon = 0.0005$, only one $E$ value (i.e., $1.48 \times 10^{-4}$) is less than $D$, while all other $E$ values are greater than $D$. Since the total number of FAKEBOB audios with four different values of $\epsilon$ is 1872, by applying our designed MEH-FEST detector, the FNR (i.e., $P_{FN}$) is only $1/1872 \approx 0.053\%$. Since $P_{FP} = 1 - P_{FN}$, the true positive rate is 99.947%. We further investigated this false-negative audio and found that it was a failed adversarial audio against the GMM SV system. Therefore, if only successful adversarial audios are considered, our detector can achieve 100% detection rate or true positive rate with the experiment set.

Furthermore, we plot the CDF of $E$ for FAKEBOB adversarial audios against i-vector SV systems with $\epsilon = 0.002$ in Fig. 7(c). It can be clearly seen that all values of $E$ of adversarial audios are greater than $D$. The smallest $E$ value is $1.55 \times 10^{-3}$. Therefore, the MEH-FEST detector can identify adversarial audios is $1.46 \times 10^{-3}$. Moreover, the average EER of five i-vector SV systems is 2.64%.

There are totally 600 original audios, including 500 audios from legitimate users and 100 audios from illegal users. We randomly selected 480 (i.e., 80%) audios as the training data and 120 audios as the test data. The minimum values of energy in the high-frequency range for the STFT of training audios (i.e., $E$) were calculated, and are with $\mu_E = 4.19 \times 10^{-5}$ and $\sigma_E = 3.68 \times 10^{-5}$. As a result, $D = 1.52 \times 10^{-4}$ based on (10). Moreover, we study the cumulative distribution function (CDF) $P(E \leq \epsilon)$, i.e., the proportion of audios that are with $E$ no greater than $\epsilon$, and plot the CDF of $E$ for training data, test data, and all original audios in Fig. 7(a). It can be seen that all three data have a similar CDF of $E$, indicating that $E$’s in these three cases have a similar probability distribution. Furthermore, we found that for the test data, the maximum value of $E$’s among 120 audios is $1.51 \times 10^{-4}$, which is less than $D$. Therefore, with our experiment set, $P_{FN}$ is zero using our proposed MEH-FEST detector, i.e., $P_{FP} = 0$. Since $P_{TN} = 1 - P_{FP}$, the true-negative rate is 1.

\begin{table}[h]
\centering
\caption{FAKEBOB Attacks Against GMM SV Systems}
\label{table:faakebob_attacks_against_gmm_sv_systems}
\begin{tabular}{| c | c | c | c | c |}
\hline
\textbf{$\epsilon$} & 0.005 & 0.002 & 0.001 & 0.0005 \\
\hline
Average ASR & 97.87% & 90.24% & 77.90% & 61.47% \\
Total running time & 17h 32m & 3h 3m & 55h 23m & 99h 5m \\
Average $\sigma$ in $T$ & 0.00366 & 0.001554 & 0.000786 & 0.000378 \\
\hline
\end{tabular}
\end{table}
all FAKEBOB adversarial audios in these i-vector SV systems, i.e., \( P_{FN} = 0 \) and \( P_{TP} = 1 \).

The experimental results indicate that our proposed MEH-FEST detector is very effective in distinguishing between original audios and FAKEBOB adversarial audios. Moreover, we found that it took the MEH-FEST detector averagely 3.37 ms to process an input audio. This shows that the MEH-FEST method is able to provide the real-time detection.

From the theoretical perspective, if (26) is applied, we found that \( \sigma_D = 2.72 \times 10^{-4} \), where \( D = 1.52 \times 10^{-4} \), \( W = 400 \), \( M = 33 \), and \( F = 492 \) from our experiments. It can be seen that all \( \sigma \)'s listed in Table I or used in the case of i-vector SV have a value greater than \( \sigma_D \). Therefore, our experimental results verify our theoretical analysis in Section IV that when \( \sigma > \sigma_D \), our designed MEH-FEST can correctly detect adversarial audios with a high probability.

## D. Performance of Countermeasures Against Adaptive FAKEBOB Attacks

We evaluate the performance of defenders' countermeasures against two adaptive FAKEBOB attacks.

1) Reducing the Perturbation Threshold \( \epsilon \): As a countermeasure, an attacker would reduce perturbation threshold \( \epsilon \) to avoid the detection of MEH-FEST. However, as shown in Table I, when \( \epsilon \) is further reduced to be less than 0.0005, the ASR will be further reduced to be less than 61.47\%, and meanwhile the running time will be further increased to be longer than 99 h 5 min. In this sense, our proposed detector can force FAKEBOB to reduce the attacking power.

We use FAKEBOB attacks with \( \epsilon = 0.00025 \) against GMM SV systems as an example. When the perturbation threshold is reduced to 0.00025, the average ASR is only 42.06\%, and the total running time increases to 143 h 40 min. On the other hand, the average \( \sigma \) in \( T \) in adversarial audios is reduced to \( 1.71 \times 10^{-4} \), which is less than \( \sigma_D \) (i.e., \( 2.72 \times 10^{-4} \)). We plot the CDF of \( E \) for FAKEBOB adversarial audios against GMM SV systems with \( \epsilon = 0.00025 \) in Fig. 8. It can be seen that a majority of \( E \) values are no more than \( D \) (i.e., \( 1.52 \times 10^{-4} \)). As a result, the FNR of the MEH-FEST method is 93.16\%. This verifies our theoretical analysis in Section IV that when \( \sigma < \sigma_D \), the adversarial audio may be able to avoid the detection of MEH-FEST.

However, adversarial audios with a small value of \( \epsilon \) are vulnerable to countermeasures such as the noise-adding method that was proposed in our previous work [3]. Such a countermeasure adds small white noise to an input audio before feeding it into an SV system. Fig. 9 shows how adding the white noise with different standard deviation can affect the normal operations of the GMM SV systems in terms of EER and ASR. Here, the adversarial audios were first generated from the GMM SV systems without the noise-adding defense and then applied to SV systems with the defense to study the ASR. It can be seen that when the standard deviation of noise increases from 0 to 0.001, EER increases from 6.20\% to only 6.63\%, while ASR decreases from 42.06\% to 7.35\%. This indicates that adding the noise with a small value of standard deviation is extremely effective in defending against FAKEBOB with a small value of the perturbation threshold \( \epsilon \), but only slightly affects the normal operations of an SV system.

2) \( n \)th FAKEBOB Attacks: We evaluate the performance of the \((n + 1)\)th MEH-FEST detection method against the \( n \)th FAKEBOB adaptive attacks. We first plot the CDFs of \( E_2 \) for training original audios, test original audios, and all original audios in Fig. 10(a). It can be seen that the probability distributions for all three cases are similar. Moreover, we also plot the CDF of \( E \) for all original audios in Fig. 10(a). Compared with \( E \), the CDF of \( E_2 \) shifts slightly to the right, indicating that \( E_2 \) has a slightly larger value than \( E \). However, the difference between \( E \) and \( E_2 \) is small. The relative small values of \( E_2 \)'s provide a foundation for the 2nd MEH-FEST method against the 1st FAKEBOB attacks.

In Fig. 10(b), the CDFs of \( E \) and \( E_2 \) for both original audios and adversarial audios from the 1st FAKEBOB attack with \( \epsilon = 0.002 \) against GMM SV systems are shown. It can be seen that the CDFs of \( E \) for the original illegal audios and adversarial audios are overlapping, indicating that the 1st FAKEBOB can effectively avoid the detection of the 1st MEH-FEST detection proposed in Algorithm 2. However, it also shows that the CDF of \( E_2 \) for adversarial audios is clearly different from that for all original audios, indicating that the 2nd MEH-FEST detection method should be able to effectively distinguish between original audios and adversarial audios.
As an extension, we further plot the CDFs of $E_{11}$ and $E_{21}$ for original audios and adversarial audios generated by the 10th or 20th FAKEBOB attacks with $\epsilon = 0.002$ against GMM SV systems in Fig. 10(c). It can be seen that there is a slight overlap between two curves of the CDFs of $E_{11}$, which means that the maximum value of $E_{11}$ from the original audios is larger than the minimum value of $E_{11}$ from the 10th FAKEBOB attack. This would introduce a nonzero FPR or FNR for our detector. Moreover, the CDF of $E_{21}$ of original audios shifts to the right of the CDF of $E_{11}$ of original audios, whereas the CDFs of $E_{11}$ and $E_{21}$ from adversarial audios are very similar in the figure. This indicates that when $n$ increases, there will be a greater overlap between two CDFs of $E_{n+1}$ from original audio and adversarial audios generated by the $n$th FAKEBOB attacks.

Table II shows the performance of the $(n + 1)$th MEH-FEST detection method against the $n$th FAKEBOB attacks with $\epsilon = 0.002$ in GMM SV systems.

### Table II

| $n$  | ASR  | Running time | $D_{n+1}$ | FPR | FNR |
|------|------|--------------|-----------|-----|-----|
| 0    | 90.24% | 31h 3m      | 0.00015   | 0%  | 0%  |
| 1    | 90.66% | 31h 54m     | 0.00017   | 0%  | 0%  |
| 10   | 90.03% | 33h 3m      | 0.00094   | 1.67% | 0%  |
| 20   | 86.81% | 37h 44m     | 0.0060    | 2.5% | 60.26% |
| 30   | 85.37% | 39h 45m     | 1.73      | 0%  | 100% |
| 40   | 81.70% | 46h 41m     | 13.22     | 2.5% | 100% |
| 50   | 76.86% | 55h 5m      | 43.65     | 5.83% | 100% |

Table III shows the performance of the $(n + 1)$th MEH-FEST detection method with a different detection threshold against the $n$th FAKEBOB attacks with $\epsilon = 0.002$ in GMM SV systems.

### Table III

| $n$  | $D_{n+1}$ | FPR | FNR |
|------|-----------|-----|-----|
| 20   | 0.0024    | 5.83% | 0.43% |
| 30   | 0.0043    | 15.83% | 17.32% |
| 40   | 0.0047    | 23.33% | 23.50% |
| 50   | 0.0057    | 34.17% | 32.48% |

However, when $n \geq 20$, while the FPR is still a small number, the FNR is very large, which indicates that the $n$th FAKEBOB attack can avoid the detection of the countermeasure.

We further manually examined the cases when $n \geq 20$ and chose a different value for the detection threshold $D_{n+1}^{\text{new}}$ as shown in Table III. It can be seen from Table III that with the new $D_{n+1}^{\text{new}}$, the FPR and the FNR of the $(n + 1)$th MEH-FEST method have a similar value, which results in a much better detection performance. How to properly choose the detection threshold for large $n$ values remains our future work.

### VII. Conclusion

In this work, we have proposed an effective detector, i.e., MEH-FEST, against FAKEBOB adversarial attacks in SV systems. The MEH-FEST detector was designed based on the observations that adversarial perturbations behave like white noise and significantly affect the audio signal when speech is absent or very weak, especially in the high-frequency range of the spectrum. Specifically, the MEH-FEST method calculates the minimum energy in the high-frequency range of the STFT of an input audio signal. We have shown through both analysis and experiments that our designed MEH-FEST is very effective in distinguish audios before and after processed by FAKEBOB attacks. Especially, we have demonstrated that both FPRs and FNRs of the MEH-FEST detector are zero or approach zero in our experiments.

Moreover, we have studied the game between attackers and defenders for the adaptive adversarial attacks and their countermeasures. Specifically, we have designed the $n$th FAKEBOB adaptive attacks that can avoid the detection of the $n$th MEH-FEST, when $n \geq m$. Meanwhile, we have shown through experiments that the $(n + 1)$th MEH-FEST method can be applied to counteract the $n$th FAKEBOB attacks, especially when $n$ is not very large.

The part of source code used in this article can be found from GitHub [33].
In this work, we have made an assumption that legitimate audio signals and illegal audio signals have a similar SNR. As our on-going work, we plan to study how to distinguish between background noise and attackers’ perturbations, so that such an assumption can be relaxed. Moreover, we plan to assess our designed MEH-FEST detector by testing it on over-the-air audio signals and measuring its performance in real-world scenarios. Furthermore, we are considering to modify our proposed MEH-FEST method to detect both replay attacks and adversarial attacks in SV systems.

REFERENCES

[1] H. Abdullah, K. Warren, V. Bindschaedler, N. Papernot, and P. Traylor, “SoK: The faults in our ASRs: An overview of attacks against automatic speech recognition and speaker identification systems,” in Proc. IEEE Symp. Security Privacy, San Francisco, CA, USA, 2021, pp. 730–747.
[2] B. Biggio et al., “Evasion attacks against machine learning at test time,” in Proc. Eur. Conf. Mach. Learn. Principles Pract. Knowl. Disc. Databases (ECML PKDD), 2013, pp. 387–402.
[3] Z. Chen, L. Chang, C. Chen, G. Wang, and Z. Bi, “Defending against FakeBob adversarial attacks in speaker verification systems with noise-adding,” Algorithms, vol. 15, no. 8, p. 293, 2022. [Online]. Available: https://doi.org/10.3390/a15080293
[4] C. Chen, X. Zhao, and M. C. Stamn, “Generative adversarial attacks against deep-learning-based camera model identification,” IEEE Trans. Inf. Forensics Security, early access, Oct. 2, 2019, doi: 10.1109/TIFS.2019.2945198.
[5] G. Chen et al., “Who is real Bob? Adversarial attacks on speaker recognition systems,” in Proc. IEEE Symp. Security Privacy, San Francisco, CA, USA, 2021, pp. 1–18.
[6] J.-M. Cheng and H.-C. Wang, “A method of estimating the equal error rate for automatic speaker verification,” in Proc. IEEE Int. Symp. Chin. Spoken Lang. Process., Hong Kong, China, 2004, pp. 265–288.
[7] N. Dalvi, P. Domingos, Mauam, S. Sanghali, and D. Verma, “Adversarial classification,” in Proc. 10th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min. (KDD), Aug. 2004, pp. 99–108.
[8] R. K. Das, X. Tian, T. Kinnunen, and H. Li, “The attacker’s perspective on automatic speaker verification: An overview,” in Proc. Interspeech, Shanghai, China, Oct. 2020, pp. 1–5.
[9] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” IEEE Trans. Audio, Speech, Lang. Process., vol. 19, no. 4, pp. 788–798, May 2011.
[10] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” Mar. 2015, arXiv:1412.6572.
[11] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, “Black-box adversarial attacks with limited queries and information,” in Proc. 35th Int. Conf. Mach. Learn., Jul. 2018, pp. 1–10.
[12] A. Jutti, C. Hsu, M. Pal, R. Peri, W. AbdAlmageed, and S. Narayanan, “Adversarial attack and defense strategies for deep speaker recognition systems,” Comput. Speech Lang., vol. 68, Jul. 2021, Art. no. 101199.
[13] S. Joshi, J. Villalba, P. Zelasko, L. Moro-Velazquez, and N. Dehak, “Study of pre-processing defenses against adversarial attacks on state-of-the-art speaker recognition systems,” IEEE Trans. Inf. Forensics Security, vol. 16, pp. 4811–4826, 2021.
[14] A. Kurakin, I. J. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” in Proc. 5th Int. Conf. Learn. Represent., Toulon, France, Apr. 2017, pp. 1–14.
[15] H. Lee. “Speech recognition.” Lecture Notes. Accessed: Sep. 2022. [Online]. Available: https://speech.ee.ntu.edu.tw/~tlkagk/courses/DLHL2020A/lec12.pdf
[16] D. Li and Q. Li, “Adversarial deep ensemble: Evasion attacks and defenses for malware detection,” IEEE Trans. Inf. Forensics Security, vol. 15, pp. 3886–3900, 2020, doi: 10.1109/TIFS.2020.3003571.
[17] J. Li, Y. Liu, T. Chen, Z. Xiao, Z. Li, and J. Wang, “Adversarial attacks and defenses on cyber–physical systems: A Survey,” IEEE Internet Things J., vol. 7, no. 6, pp. 5103–5115, Jun. 2020.
[18] X. Li et al., “Investigating robustness of adversarial samples detection for automatic speaker verification,” in Proc. Interspeech, Shanghai, China, Oct. 2020, pp. 1–5.
[19] D. Lowe and C. Meer, “Adversarial learning,” in Proc. 11th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min., Aug. 2005, pp. 641–647.
[20] J. H. McClellan, R. W. Schafer, and M. A. Yoder, DSP First, 2nd ed. London, U.K.: Pearson Educ., Inc., 2016.
[21] M. Muller, Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications. New York, NY, USA: Springer, 2015.
[22] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “LibriSpeech: An ASR corpus based on public domain audio books,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP), South Brisbane, QLD, Australia, Apr. 2015, pp. 5206–5210.
[23] D. Povey et al., “The Kaldi speech recognition toolkit,” in Proc. IEEE Workshop Autom. Speech Recogn. Understanding, Dec. 2011, pp. 1–4.
[24] A. Rahman, M. S. Hossain, N. A. Alrajeh, and F. Aisolami, “Adversarial examples—Security threats to COVID-19 deep learning systems in medical IoT devices,” IEEE Internet Things J., vol. 8, no. 12, pp. 9603–9610, Jun. 2021.
[25] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, “Speaker verification using adapted Gaussian mixture models,” Digital Signal Process., vol. 10, nos. 1–3, pp. 19–41, Jan. 2000.
[26] C. Szegedy et al., “Intriguing properties of neural networks,” in Proc. Int. Conf. Learn. Represent., Apr. 2014, pp. 1–10.
[27] W. Tang, B. Li, S. Tan, M. Barni, and J. Huang, “CNN-based adversarial embedding for image steganography,” IEEE Trans. Inf. Forensics Security, vol. 14, pp. 2074–2087, 2019, doi: 10.1109/TIFS.2019.2891237.
[28] F. Tramer, N. Carlini, W. Brendel, and A. Madry, “On adaptive attacks to adversarial example detection,” in Proc. Conf. Neurol. Inf. Process. Syst. (NeurIPS), 2020, pp. 1–13.
[29] J. Villalba, S. Joshi, P. Zelasko, and N. Dehak, “Representation learning to classify and detect adversarial attacks against speaker and speech recognition systems,” in Proc. Interspeech, 2021, pp. 1–5.
[30] 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 Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS), Oct. 2020, pp. 1–17.
[31] H. Wu, Y. Zhang, Z. Wu, D. Wang, and H. Lee, “Voting for the right answer: Adversarial defense for speaker verification,” in Proc. Interspeech, 2021, pp. 1–5.
[32] Z. Yang, B. Li, P. Chen, and D. Song, “Characterizing audio adversarial examples using temporal dependency,” in Proc. Int. Conf. Learn. Represent., 2019, pp. 1–15.
[33] "Detection_FAKEBOB." GitHub. Accessed: Sep. 2022. [Online]. Available: https://github.com/zeshengchen/detection_FAKEBOB
[34] "Google cloud platform." Google. Accessed: Sep. 2022. [Online]. Available: https://cloud.google.com/
[35] "Hann window, SciPy documentation." Accessed: Sep. 2022. [Online]. Available: https://docs.scipy.org/doc/scipy-1.0.0/reference/generated/scipy.signal.hann.html
[36] W. Kenton. “Three sigma limits statistical calculation, with an example." Investopedia. Accessed: Mar. 2023. [Online]. Available: https://www.investopedia.com/terms/b/threesigmalimits.asp
[37] "A Python package for music and audio analysis." Librosa. Accessed: Sep. 2022. [Online]. Available: https://librosa.org/
[38] "Audio signal." Wikipedia. Accessed: Sep. 2022. [Online]. Available: https://en.wikipedia.org/wiki/Audio_signal
[39] "Chi-squared distribution." Wikipedia. Accessed: Sep. 2022. [Online]. Available: https://en.wikipedia.org/wiki/Chi_squared_distribution
[40] "Complex normal distribution." Wikipedia Accessed: Sep. 2022. [Online]. Available: https://en.wikipedia.org/wiki/Complex_normal_distribution
[41] "Monte Carlo method." Wikipedia. Accessed: Sep. 2022. [Online]. Available: https://en.wikipedia.org/wiki/Monte_Carlo_method
[42] "Normal distribution." Wikipedia. Accessed: Mar. 2023. [Online]. Available: http://www.wikipedia.org/wiki/Normal_distribution
[43] "Repeated game." Wikipedia. Accessed: Mar. 2023. [Online]. Available: https://en.wikipedia.org/wiki/Repeated_game

Zesheng Chen (Senior Member, IEEE) received the M.S. and Ph.D. degrees from Georgia Institute of Technology, Atlanta, GA, USA, in 2005 and 2007, respectively. He is currently an Associate Professor with the Department of Computer Science, Purdue University Fort Wayne, Fort Wayne, IN, USA. His current research interests include machine learning security, network security, Internet of Things, and performance evaluation.