Inference Attacks for X-Vector Speaker Anonymization

Luke A. Bauer
University of Florida
lukedrebauer@ufl.edu

Wenxuan Bao
University of Florida
wenxuanbao@ufl.edu

Malvika Jadhav
University of Florida
jadhav.m@ufl.edu

Vincent Bindschaedler
University of Florida
vbindschaedler@ufl.edu

***Abstract*—We revisit the privacy-utility tradeoff of x-vector speaker anonymization. Existing approaches quantify privacy through training complex speaker verification or identification models that are later used as attacks. Instead, we propose a novel inference attack for de-anonymization. Our attack is simple and ML-free yet we show experimentally that it outperforms existing approaches.**

***Index Terms*—Speaker Anonymization, Inference Attacks, Machine Learning**

1. Introduction

The past decade has spurred major advances in user-facing speech processing technologies. This include Automatic Speech Recognition (ASR) [1], text-to-speech (TTS) synthesis [2], and speaker identification or verification [3]. Another technology that has received recent attention, especially in academic work is speaker anonymization [4], [5], [6], [7]. The goal of speaker anonymization is to transform speech samples to obscure the identity of the speakers while preserving the content.

A straightforward realization of a speaker anonymization system is to use an ASR model to transcribe the input speech and then synthesize a new audio sample matching the transcription but with a different voice. This approach perfectly protects the privacy of speakers and the resulting audio samples are still useful for some downstream tasks due to preserving the transcripts. However, it sacrifices naturalness in the synthesized voice and discards all of the prosodic features (pitch, emotionality, tone, etc.) present in the original speech samples.

To ensure that we can preserve naturalness and prosody in speech while anonymizing speakers, a number of approaches have been proposed [4], [5], [6], [8], [7]. Notably, initiatives such as the VoicePrivacy Challenges [6], [9], [8], [7] have been encouraging development and evaluation of speaker anonymization techniques. One particularly prominent approach is the use of x-vectors [10], which are embeddings that capture speaker-specific characteristics extracted from speech. Informally, x-vector anonymization schemes transform this x-vector into a *pseudo x-vector* using publicly available data (i.e., a public pool of other speakers' x-vectors). This pseudo x-vector is then combined with the transcript and pitch information (extracted from the input speech sample) to synthesize a new audio sample. When the pseudo-x-vector is chosen to be significantly different from the original speaker's x-vector, the synthesized speech sounds natural, as if it was uttered by another person.

However, despite substantial research, the privacy-utility tradeoff of x-vector speaker anonymization is not fully characterized. This is partially due to the fact that previous work has primarily relied on empirical privacy evaluation using speaker identification or speaker verification systems. Said differently, existing approaches train deep learning models to play the role of the attacker.

Training machine learning models as attacks has proven successful in other contexts [11], [12]. However, for speaker anonymization, we argue that using a model to identify relevant patterns in speech to de-anonymize speakers is not the best strategy. A better strategy is to design principled inference attacks and use those as lower bounds to quantify privacy. To demonstrate this, we propose a simple (ML-free) de-anonymization attack that leverages the specifics of the transformation of the original x-vector into the pseudo x-vector. We show empirically that this attack significantly outperforms current ML-based approaches based on training speaker verification/identification models.

Our proposed attack is not only simpler and more accurate than existing alternatives, it is also more computationally efficient as it does not require training any machine learning models. Moreover, our attack is able to detect if the target speaker is not within the set of considered suspects, so it still infers information in such cases. Our results call for re-aligning evaluation of the privacy-utility tradeoff for x-vector speaker anonymization. Machine learning is a powerful, but it should *not* be used as the sole strategy for analyzing privacy-utility tradeoffs.

2. Background & Related Work

2.1. VoicePrivacy Challenge

As voice-based technologies become pervasive [13], [14], concerns about protecting personal information embedded in speech [15] have become increasingly urgent. In response, the VoicePrivacy Challenge [9], [7], [8] offers a platform for researchers to explore and compare state-of-the-art methods to protect a speaker's identity while preserving

critical linguistic content.[1] In the VoicePrivacy Challenge, anonymization techniques are evaluated from both privacy and utility perspectives. Typically privacy performance is evaluated using the Equal Error Rate (EER) of a speaker verification/recognition model, which indicates how successfully a method prevents speaker re-identification, whereas utility is evaluated using Word Error Rate (WER) and Unweighted Average Recall (UAR) comparing the anonymized speech transcript to the original speech transcript. The VoicePrivacy Challenge also provides several baseline systems to guide researchers, the most popular of which is the class of x-vector speaker anonymization approaches [16], [17], [18].

2.2. X-Vector Speaker Anonymization

A x-vector based speaker anonymization system [9], [7] first extracts x-vectors, fundamental frequencies, and bottleneck features from original speech. It then anonymizes the speaker's identity by replacing the original x-vector with an average of numerous vectors selected from the public x-vector pool, creating a *pseudo-speaker x-vector*. Finally, using a neural source-filter model, it synthesizes a new speech waveform that retains the original's linguistic content but sounds as though it was uttered by a different individual.

Fang et al. [4] introduced the first x-vector based speaker anonymization method through voice conversion, which adapts the x-vector of a speaker to match a target x-vector. Following this, a number of other strategies were proposed [19], [20], [21] that include random modification of the embeddings, Singular Value Decomposition, and Wasserstein GAN to generate the target x-vector.

Champion et al. [22] performed linkability and invertibility attacks on anonymized x-vectors produced using the baseline system of the VoicePrivacy challenge [23]. This work used two different embedding alignment algorithms to evaluate x-vector based anonymization in scenarios where the attacker was completely informed or semi-informed about the original x-vector and its corresponding anonymized x-vector. Unlike ours, they use a machine learning based method. Champion et al. [24] also analyzed x-vector based speaker anonymization proposed by [4] where the attacker has complete knowledge of the system.

There are different x-vector based speaker anonymization techniques. In this paper, we use baseline B1 in the 2024 VoicePrivacy Challenge (which is baseline B1.b in the 2022 challenge). Fig. 1 shows its architecture. We consider it as our main representative x-vector anonymization technique.

2.3. Privacy Evaluation

Privacy evaluation in the VoicePrivacy Challenge [9], [7], [8] assumes the attacker trains an automatic speaker verification (ASV) model on anonymized data. For each speaker, the attacker computes an average embedding from all anonymized enrollment utterances and compares it to the embedding of an anonymized trial utterance to verify speaker identity. They use EER as evaluation metrics. Srivastava et al. [17] evaluates privacy by assessing how well the speaker anonymization methods prevent an attacker from re-identifying the speaker using automatic speech recognition (ASR) model. The privacy evaluation is quantitatively measured using linkability scores, which reflect the likelihood of correctly linking anonymized speech to the original speaker. It outlines various attacker knowledge levels, ranging from ignorant (unaware of anonymization) to semi-informed (aware of some details of the anonymization method but not others), influencing how the attacker might use the anonymized data to re-identify the speaker. Champion et al. [24] also evaluate the privacy of x-vector based speaker anonymization but in a white-box setting when the target selection is restricted to a specific identity. The privacy evaluation is performed using the linkability metric with an x-vector-PLDA based Automatic Speaker Verification (ASV) system from the VoicePrivacy Challenge.

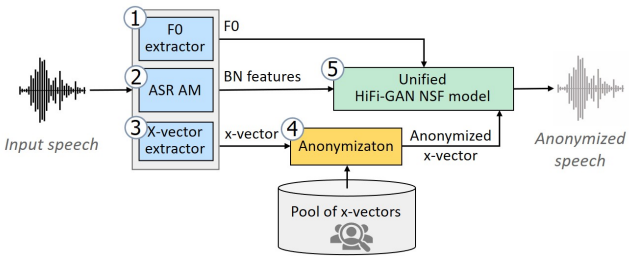

Figure 1: Model architecture for x-vector speaker anonymization.[2]

In this paper, we build on the concepts presented in [16], [7] but propose a novel privacy attack by examining a more knowledgeable adversary than has been considered in related work, distinct from the framework used in [17]. Furthermore, our main insight is that x-vector speaker anonymization can be attacked directly by leveraging how they construct the pseudo x-vector from the original speech. It is not necessary to train a complex ASI/ASV model and hope it learns to de-anonymize.

3. Problem Statement

We frame the problem of speaker anonymization as follows. The input is a speech sample a from a speaker s whose identity we aim to protect. We use a speaker anonymization method that produces an anonymized speech sample y such that y approximately preserves the transcript of a but obscures speaker identity. Some features, such as gender, may still be identifiable, but the speaker should be indistinguishable from other speakers that share those features.[3] The adversary observes the anonymized speech

1. https://www.voiceprivacychallenge.org/

2. Image source: https://github.com/Voice-Privacy-Challenge/Voice-Privacy-Challenge-2022/blob/master/baseline/fig/B1b.jpg (License: Creative Commons Attribution 4.0 International)

3. Some methods preserve gender by restricting the public pool of x-vectors used to construct pseudo-speaker x-vector to the same gender as the original speaker.

sample y and attempts to determine the identity of the speaker, i.e., de-anonymize (re-identify) them.

3.1. Formalizing Embedding-based Anonymization

We think of embeddings being x-vectors but the formalization extends to other (potentially future) representations of an individual's voice features. An Embedding-based Anonymization Scheme is a tuple $(\text{Ext}, \text{Transf}, \text{Synth})$ of algorithms where:

- $\text{Ext}(a) \to (x, t)$: Extract takes as input a speech sample a and outputs a text transcript t (a natural language string) and a speaker embedding $x \in \mathbb{R}^k$ (the x-vector) where k is the embedding dimension (e.g., $k = 512$).
- $\text{Transf}(x) \to p$: Transform takes as input a speaker embedding x and transforms it (anonymizes it) into a different pseudo speaker embedding p.
- $\text{Synth}(t, p) \to y$: Synthesize takes as input a text transcript t and a pseudo speaker embedding p and produces an audio speech sample y as output.

This captures the idea and anonymization process of existing x-vector speaker anonymization schemes such as [6], [9], [7]. That is, given an audio sample a to anonymize, we use the function $\text{Ext}(a)$ to extract its transcript t and x-vector x, then transform this x-vector into a *pseudo* x-vector p using $\text{Transf}(x)$, and then finally use the function $\text{Synth}(t, p)$ to synthesize a new audio sample that matches the characteristics of the speaker represented by p.

There are a few important remarks. Since x-vector transformations rely on a public pool of embeddings, we can think of Transf as including this pool implicitly. To capture any randomness in the process of transforming an x-vector into another *pseudo* x-vector, we can think of Transf as having an auxiliary input r which is a source of randomness. For the purposes of thinking about formal security, we can even assume that r is derived (for each invocation) from a cryptographic secret key, which is equivalent to thinking of the randomness in Transf as coming from a PRNG seeded with the secret key. To simplify the presentation we omit this from the description and view Transf as probabilistic.

In our representative scheme, the VoicePrivacy Challenge's x-vector anonymization implementation, the extract function actually produces a tuple (x, F_0, B_N) where x is the x-vector (embedding), F_0 is the pitch of the speaker, and B_N represents the features of the transcript. Further, the synthesis function Synth takes as input an pseudo x-vector p in addition to F_0 and B_N, and therefore it is implicitly assumed that *no information about speaker identity* is contained in F_0 and B_N. Consistent with related work Shamsabadi et al. [5] we found that this is assumption is false empirically. We discuss this in Section 5.6.

3.2. Privacy

A natural way to perform a de-anonymization attack is to try to (approximately) invert the transformation $\text{Transf}(\cdot)$. If we observe an anonymized speech sample with some pseudo-speaker embedding p, we want to identify the most likely speaker embedding x such that $\text{Transf}(x) \to p$. This can be viewed as an inference attack where the attacker guesses based on some likelihood ratio or approximates it. For example, suppose we have x-vectors x_1 and x_2 from two distinct possible speakers and an (estimated) pseudo x-vector p obtained from the anonymized audio. The attack computes ratio: $\Pr(\text{Transf}(x_1) = p)/\Pr(\text{Transf}(x_2) = p)$. The attacker guesses the first speaker if and only if the ratio is larger than 1. The challenge is therefore to compute or approximate the ratio. This is the approach of the attack we propose in Section 4, although we avoid explicitly computing probabilities and use x-vector distances instead.

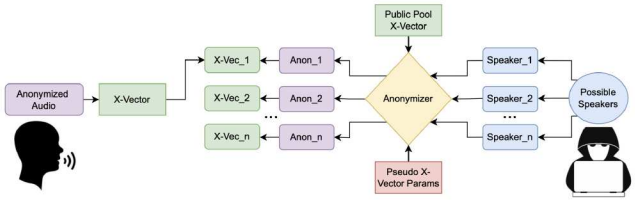

Figure 2: Illustration of our proposed attack. The target speaker, s, has released anonymized audio that the adversary is attempting to identify. The adversary has a set of potential speakers, S' and they believe the speaker is a part of it. They anonymize audio from these potential speakers. Finally, they extract an x-vectors from the target anonymized audio x and from the set of anonymized audio they just generated X'. The speaker in S' whose x-vector most closely matches x is the speaker of the target audio.

By contrast, the approach taken in the VoicePrivacy challenges [6], [9], [7] and related work is to perform de-anonymization by training an Automatic Speaker Verification (ASV) model. This model is then fed the anonymized speech sample and asked to predict the identity of the speaker. This is how various methods for the challenge are evaluated and compared to each other in terms of privacy. There are a number of downsides with this ML-based approach. In particular, it does not utilize fine-grained information about the anonymization method (Transf) or its invertibility. It requires training the ASV model which is computationally intensive and the performance of the attack depends on how well that model can learn speaker identity from (anonymized) speech samples.

3.3. Utility

Privacy is not the only important criterion for speaker anonymization. Otherwise we could simply transcribe the original speech sample and then synthesize a new audio sample for that transcript with a fixed voice (e.g., robotic voice) independent of the original speaker. However, this would not preserve features such as gender, age [25], emotion [26], or tone/speaking style, etc.

In the literature and the VoicePrivacy challenge, utility is often measured as distortion to the transcript (differences in transcription from the original to the anonymized audio), which are measured using WER (Word Error Rate). We

also measure FAD (Fréchet Audio Distance) [27], which compares generated audio against a ground truth set of non-generated audio to determine quality.

4. Methodology

4.1. VoicePrivacy Challenge

Consider how the VoicePrivacy Challenge anonymizes audio [6], [7]. The system first extracts an x-vector, and other data, such as pitch (F_0) and linguistic features (B_N), from the original audio. It then creates a *pseudo x-vector* which is given, along with the F_0 and B_N collected from the original audio, to a Neural Source Filter model. The model uses these to create the final anonymized audio.

The system uses a public pool of x-vectors, which we denote as pool. This pool is constructed from a large number of x-vectors gathered from independent audio. The VoicePrivacy Challenge uses the "train-other-500" subset of LibriTTS [28] as the source of this pool. There are thousands of utterances from about 500 speakers. Finally, the pool is divided by gender, which provides the ability to use either the same or opposite gender pool as the original speaker. In our experiments, we assume same gender pools.

To construct a pseudo x-vector, the affinity between the original audio's x-vector and every other vector in the gender filtered pool is calculated. Affinity is a distance between vectors, and can be calculated either using the cosine distance or using PLDA. The list of affinity values is then sorted and then the top 200 vectors are selected. This may be the top 200 nearest vectors (highest affinity) or the 200 farthest vectors (lowest affinity) from the original x-vector. From there a subset of 100 vectors are randomly selected, and then averaged together. This average vector is the *pseudo x-vector*. This pseudo x-vector is then applied at either the speaker level (all utterances of a speaker get the same x-vector) or the utterance (each utterance gets its own x-vector). We consider speaker-level for our experiments. Finally, the pseudo x-vector is provided as input to the generation model, along with F_0 and B_N, to synthesize the anonymized audio.

4.2. Proposed Inference Attack

To reiterate, a speaker s has several audio files a they wish to anonymize. For example, these samples may consist of the speaker reading several sentences from a book, each sentence being a separate audio file also described as an "utterance". The speaker anonymizes a to get new anonymized audio y. The adversary observes the anonymized speech sample y and attempts to guess the identity of the speaker. The adversary has access to the anonymization method, as well as a set of potential speakers S', for which they have audio samples A'. They believe that s is within S', and they attempt to identify them.

Our proposed inference attack works as follows. First, the attacker extracts an x-vector x from the observed anonymized speech sample y. This is done using the x-vector extractor from the VoicePrivacy Challenge, based on Snyder et al. [29]. In principle, this extracted x-vector should be similar, if not identical, to the pseudo x-vector used to create it. In practice, there are significant differences due to the audio generation and x-vector retrieval process. We will discuss this problem later. The attacker then simulates the anonymization process for each speaker in S'. From this, the attacker obtains anonymized audio for each speaker in S' from which they extract x-vectors. This yields a set X' of x-vectors where each x-vector $x'_i \in X'$ represents the speaker embedding/identity of s'_i. Finally, the attacker compares each x'_i to x to find the speaker most similar to the original target s (using l_2 distance). Algorithm 1 shows details of the attack method.

Algorithm 1 De-Anonymization Attack

Input: y: Anonymized audio files from target speaker s;
pool: Public pool of x-vectors;
S': Set of potential speakers;
A': Original utterances from speakers in S'
Output: s'_i in S' that is most likely to be s

procedure EXTRACT_XVECTOR(audio)
    Given several samples of audio from a speaker extract an x-vector for that speaker.
end procedure
procedure ANONYMIZE(audio, pool)
    Run representative x-vector anonymization method to obtain anonymized audio.
end procedure
$x \leftarrow$ Extract_xvector(y)
for s'_i in S' **do**
    $y'_i \leftarrow$ Anonymize(a'_i, pool)
    $x'_i \leftarrow$ Extract_xvector(y'_i)
    $\text{dist}_i \leftarrow ||x'_i - x||_2$
end for
return s'_i with the lowest dist

This attack works by leveraging the pseudo x-vector construction method. One may expect the anonymized audio to contain no information about the identity of the original speaker, since it is constructed from the pseudo x-vector. However, the pseudo x-vector is not constructed independently from the original speaker x-vector, thus it carries statistical information from it. The specific steps of selecting the 200 nearest/farthest x-vectors (to the original speaker x-vector) from the pool leaks information about it through the affinity/distance. In the ideal case for the adversary, the 200 nearest/farthest x-vectors acts as fingerprint for the speaker identity s. In such cases, the inference attacks only needs to overcome the uncertainty of the 100 randomly selected x-vectors from within the 200 nearest/farthest x-vector set.

5. Experiments

5.1. Setup

To evaluate our method we apply the VoicePrivacy Challenge anonymization process to the Libri_Dev dataset [30].

TABLE 1: Attack accuracy and utility for different pseudo x-vector construction methods. We show accuracy for our main attack method under the Same and Different adversary models, as well as an ASI model using Same, Different, and Original adversary models. For utility, we show the WER and FAD of all pseudo x-vector methods. Random Single has the lowest attack accuracy and best utility.

	(Ours) Same	(Ours) Different	ASI Same	ASI Diff	ASI Orig	WER	FAD
Original Audio	100	96.7	79.3	70.1	79.3	5.4	2.1
200 Farthest	100	76.3	34.5	19.6	9.2	7.4	7.5
200 Nearest	100	77.6	43.6	20.2	12.6	7.8	7.2
50 Farthest	100	65.7	32.4	18.2	11.1	7.7	7.5
50 Nearest	100	73.9	31.5	17.8	14.7	7.6	6.9
Random Average	100	46.3	36.3	18	9.8	6.4	7.4
Random Single	20	11.2	33.2	14.6	5.3	6.9	6.5

TABLE 2: Pseudo x-vector construction methods. All consist of selecting a *World* of nearest or farthest x-vectors from pool, then averaging a random subset of them together. Random Single is the exception since it only uses a single x-vector instead of an average.

Method	Description
200 Farthest	Average 100 vectors out of 200 farthest
200 Nearest	Average 100 vectors out of 200 nearest
50 Farthest	Average 25 vectors out of 50 farthest
50 Nearest	Average 25 vectors out of 50 nearest
Random Average	Average 100 vectors of the entire pool
Random Single	A single vector from the entirety of pool

`Libri_dev_trials_m/f` are considered the original pool of audio A belonging to speakers S, which is then anonymized. `Libri_dev_enrolls` is used as the adversary's target pool, S' when a separate and distinct pool is required. There are 29 unique speakers in S', which is the number of unique speakers in `Libri_dev_enrolls`. For some experiments we use a smaller subset to show how the size of S' influences attack accuracy. Each speaker s_i is associated with a_i, composed of several utterances of the speaker reading sentences from a book. The exact number of utterances depends on the speaker.

We wish to examine how the pseudo x-vector generation parameters affect privacy and utility. We discussed how the VoicePrivacy Challenge constructs pseudo x-vectors in Section 4, by gathering a *World* of 200 nearest / farthest x-vectors from which 100 are randomly selected. We consider both settings to evaluate our representative method. We further evaluate a smaller *World* of 50 from which 25 are sampled, to show how the size of the pool potentially influences the privacy of the speakers. Finally, we also evaluate two additional scenarios that should provide maximum privacy. The first takes the average of 100 randomly selected x-vectors from *the entire pool* and uses it as the pseudo x-vector. The second selects a single x-vector from the pool randomly and uses it as the pseudo x-vector. Since the selection of pseudo x-vector in those two scenarios is not dependent on the original speaker or speech, there should be no leakage.

5.2. Attack Scenarios

Recall that in our threat model the adversary knows the anonymization method, the pseudo x-vector construction method, and any other parameters. As a result, the adversary can replicate steps taken during the anonymization process, such as x-vector extraction and pseudo x-vector construction. The adversary also has access to a set of potential speakers S', for whom the adversary has original non-anonymized audio samples A'.

TABLE 3: Different adversary knowledge levels we evaluate. We evaluate Same, Different, and Unknown for our attack. We evaluate Same, Different, and Original for the ASI model.

Knowledge	Description
Same	The target's original non-anonymized audio is in A'
Different	The target speaker s is in S' but with different utterances
Original	(ASI only) Original non-anonymized audio to train model

In Table 3 we propose three attack scenarios that map onto different adversarial knowledge levels, representing how closely the adversary's audio samples mirror the samples used by the anonymization method. Our most powerful attack assumes that the adversary knows the original speaker is within that set and that the audio samples are the same as the ones used to generate the target audio y, i.e. $A' = A$. We also evaluate a weaker, but perhaps more realistic scenario, where S' contains the target speaker, but with different audio samples, i,e, $A' \neq A$. For the ASI model, we also evaluate the scenario where the adversary does not have any anonymized audio and trains only on the original audio. We further evaluate the scenario where the adversary does not know if the target speaker is within S'.

Attacks. We evaluated our proposed inference attack, implemented as described in Section 4. As a comparison point, we use an Automatic Speaker Identification (ASI)-based attack, which attempts to identify the original speaker by examining anonymized audio samples rather than the x-vectors. More specifically, we train an ASI neural network [31] using data corresponding to the assumed adversary knowledge levels and perform inference to re-identify the target speakers. We train until loss stops decreasing, usually around 10 epochs. We use the `Libri_dev` male and female subsets as training samples. Related work often relies on an ASV (which takes in audio files and a claimed identity to verify the speaker) for this. Instead, we opted to use a speaker identification model since it better fits our attack scenario, having a set of audio files and attempting to classify each one as a certain speaker.

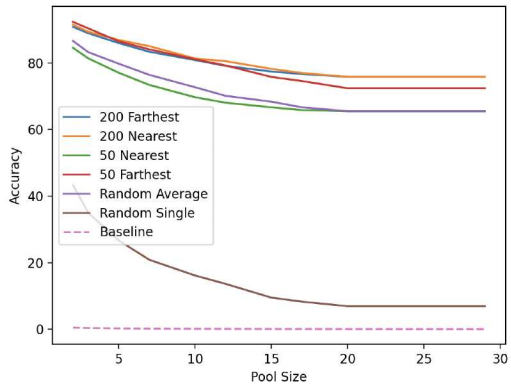

Figure 3: Attack accuracy for different pseudo x-vector construction methods under the Different adversary knowledge level. All construction methods, except for random, perform much better than random guessing. As the adversary's pool size increases, the attack accuracy decreases before leveling out around a pool of size 20.

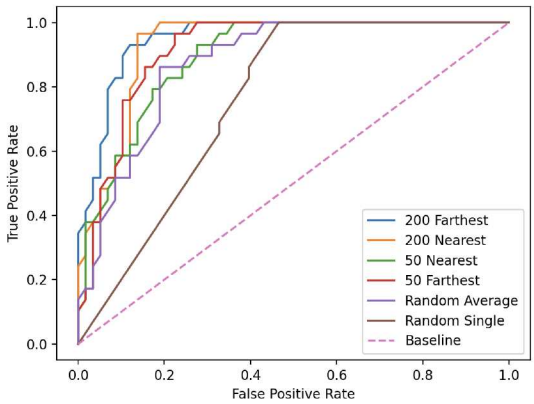

Figure 4: ROC curves for identifying if a speaker is within the potential target pool when anonymized using different pseudo x-vector construction methods. All results shown are under the Different adversary knowledge level. AUC is far above random guessing for all. Not shown is the Same adversary knowledge level, which has AUC=1 for all pseudo x-vector construction methods, except Random Single, which is still above random guessing.

5.3. Results

Table 1 gives the results for our attack at different adversary knowledge levels and pseudo x-vector construction methods. We find that at the Same adversary knowledge level, we are able to achieve perfect de-anonymization accuracy at all sizes of S' and for (nearly) all generation parameters. Random Average and Random Single achieve better accuracy than expected, due to leakage of F_0 and B_N features in the x-vector extraction process, which we further discuss later on.

Results for the Different adversary knowledge scenario are shown in Fig. 3. Arguably this is a more realistic adversary scenario, and is reflective of the most knowledgeable adversary considered in related work. Nevertheless, our inference attack still achieves high re-identification accuracy.

Table 1 also shows the results for the ASI at various adversarial knowledge levels. As expected the model performs better with stronger adversaries. Perhaps, due to the small sample sizes for each speaker in `Libri_dev`, it is unable to achieve high accuracy.

Overall results suggest that training complex deep neural networks as an attack is unnecessary, as our (ML-free) inference attacks performs as well or better without training any model. It only requires knowledge of the anonymization method and the public pool of x-vectors. Furthermore, given the high success rate, the anonymization methods considered do not appear to provide meaningful privacy.

5.4. Open World Evaluation

We also evaluated the open world setting where it is not known if the original speaker s is within the set S' of possible targets. The question in this case is whether the attack is resilient to the possibility that the original speaker is not in the target set.

To evaluate this, we use our inference attack, but we only include the target speaker s within S' with probability 0.5 each time. Our attack accounts for this as follows. Instead of sorting possible speaker by their distance to the x-vector uses to construct the anonymized audio, the attack compares the minimum distance against a threshold. If the distance is smaller than the threshold then the adversary will declare the target is within the set. To determine a suitable threshold, the adversary can perform the attack in a setting where the speaker s is within the set S' to estimate the distribution of x-vector l_2 distances and then derive a threshold value from it (e.g., choosing a threshold at or above a given percentile).

We found that in the Same adversary knowledge setting we again get perfect accuracy, i.e., we are able to select a threshold without any false positives or false negatives. Fig. 4 shows the ROC curves for the Different adversary knowledge scenario. We observe that even in this case, the adversary is able to identify if the speaker is present with high accuracy.

5.5. Other Experimental Results

Time Comparison. Anonymizing `Libri_dev` took approximately 21 minutes, most of it to extract all the necessary information and generate the anonymized audio. Note that both attacks methods (ours and ASI/ASV) must perform this step. However, from there it took approximately 12 minutes to train the ASI model to convergence, solely on `Libri_dev`. Note that the ASV used in the VoicePrivacy Challenge can take up to 10 hours to train on their recommended dataset. However, the only time consuming step in our attack is extracting x-vectors from `Libri_dev`, which only takes approximately 2 minutes. Our attack is thus much faster, and needs no training or additional models.

Utility. To evaluate how the different generation parameters influenced utility, we compare anonymized audio against

non-anonymized audio using WER and FAD. While utility is maintained across methods, Random Single has high utility across both metrics. Our results suggest Random Single is the best option, since it also achieves the highest privacy across all knowledge levels.

TABLE 4: Results for our normalized scenario, where the generator is given the same F0 and BN regardless of the input audio. Hence the only difference between audio files is the x-vector used to generate it. We also show the accuracy for our Same and Different knowledge level attacks for comparison.

	Same	Different	Normalized
Original Audio	100	96.7	100
200 Farthest	100	76.3	93
200 Nearest	100	77.6	93.1
50 Farthest	100	65.7	79.3
50 Nearest	100	73.9	99.6
Random Average	100	46.3	3.5
Random Single	20	11.2	3.4

5.6. Idiosyncrasies of X-Vector Anonymization

Recall that Random Single and Random Average should have no privacy leakage. Yet, the results (Table 1) show that our attack on these methods achieve well above random guessing accuracy. This should not happen because the pseudo x-vectors should contain no information about the original speaker. We discovered that the reason for the empirical outperformance is leakage of information about F_0 and B_N into the anonymized audio. We believe this was also observed indirectly by Shamsabadi et al. [5].

To evaluate how this affects our attack success, we performed the attack again while forcibly setting the F_0 and B_N features to ensure no leakage from them. This means both y and y' will have the same utterances, pitch values, and parameters, except for the pseudo x-vector used to generate them. We call this the *normalized* scenario. Results are shown in Table 4 where we see that (as expected) both Random Average and Random Single achieve only random guessing accuracy. Recall the size of S' is based on `Libri_dev_enrolls`, which contains 29 unique speakers. Therefore, random guessing accuracy is 3.45%. Our attack on other methods also sees small decreases to accuracy, but can still easily identify the speaker most of the time.

6. Discussions & Limitations

Discussion. Although not the focus of our paper, we found empirically that is surprisingly easy to identify audio that has been anonymized with x-vector based methods when compared against non-anonymized audio. We fine-tuned a Whisper [32] model to distinguish between audio anonymized with 200 farthest pseudo x-vectors and non-anonymized audio samples from the same dataset and found it easily reached perfect accuracy. This is likely due to a combination of averaged pseudo x-vectors resulting in very neutral sounding audio. We believe this is noteworthy, especially in light of the recent attention on deepfake audio [33], [34]. However, it is unlikely to be major issue for speaker anonymization except in scenarios where concealing that anonymization has taken place is essential.

Our results suggest that any x-vector anonymization that carries information from the original x-vector to the pseudo x-vector can be broken. However, preserving features such as tone, pitch, gender, etc. may be essential for utility. This is why approaches that achieve stronger privacy such as those based on differential privacy [5], [35] or methods such as Random Single and Random Average may not be practical in many scenarios. Random Single, in particular, also has the potential drawback that it selects the x-vector of an actual speaker from the pool and thereby essentially results in impersonating that individual.

Limitations. Our machine learning experiments using speaker identification (ASI) to classify audio as belonging to one of several speaker. This makes sense in our setting. Nevertheless it would be appropriate in future work to consider the case of speaker verification (ASV) for the attack model, where a model is trained to recognize audio as belonging to a single speaker.

Another limitation is scalability, which we have not evaluated, especially in cases where the pool contains a very large number (e.g., millions) of distinct speakers. This is particularly challenging since our attack needs to compare the distances of all potential targets against the pool, which quickly becomes prohibitive as the pool size grows. Future work may consider optimizations to narrow down the potential pool, or speed up comparisons.

7. Conclusions & Future Directions

We proposed a novel attack on x-vector speaker anonymization that does not require any additional model training. The existence of this attack and its outperformance of existing automated speaker identification attack approaches underlines the importance of considering the specifics of the anonymization process, and not just the end result audio hoping that attack models will learn to identify relevant patterns during training.

Future research can build on our results by taking into account the invertibility of pseudo x-vector generation methods to optimize the privacy-utility tradeoff. Additional assessments of utility through user study experiments could further improve pseudo x-vector construction methods.

Acknowledgments

This work was supported in part by the National Science Foundation under CNS-1933208. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] D. Yu and L. Deng, *Automatic speech recognition*. Springer, 2016, vol. 1.

[2] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," *Advances in neural information processing systems*, vol. 32, 2019.

[3] R. Togneri and D. Pullella, "An overview of speaker identification: Accuracy and robustness issues," *IEEE circuits and systems magazine*, vol. 11, no. 2, pp. 23–61, 2011.

[4] F. Fang, X. Wang, J. Yamagishi, I. Echizen, M. Todisco, N. Evans, and J.-F. Bonastre, "Speaker anonymization using x-vector and neural waveform models," *arXiv preprint arXiv:1905.13561*, 2019.

[5] A. S. Shamsabadi, B. M. L. Srivastava, A. Bellet, N. Vauquier, E. Vincent, M. Maouche, M. Tommasi, and N. Papernot, "Differentially private speaker anonymization," *arXiv preprint arXiv:2202.11823*, 2022.

[6] N. Tomashenko, B. M. L. Srivastava, X. Wang, E. Vincent, A. Nautsch, J. Yamagishi, N. Evans, J. Patino, J.-F. Bonastre, P.-G. Noé *et al.*, "The voiceprivacy 2020 challenge evaluation plan," Ph.D. dissertation, LIA-Laboratoire Informatique d'Avignon; MULTISPEECH-Speech Modeling for ..., 2020.

[7] N. Tomashenko, X. Miao, P. Champion, S. Meyer, X. Wang, E. Vincent, M. Panariello, N. Evans, J. Yamagishi, and M. Todisco, "The voiceprivacy 2024 challenge evaluation plan," *arXiv preprint arXiv:2404.02677*, 2024.

[8] M. Panariello, N. Tomashenko, X. Wang, X. Miao, P. Champion, H. Nourtel, M. Todisco, N. Evans, E. Vincent, and J. Yamagishi, "The voiceprivacy 2022 challenge: Progress and perspectives in voice anonymisation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[9] N. Tomashenko, X. Wang, E. Vincent, J. Patino, B. M. L. Srivastava, P.-G. Noé, A. Nautsch, N. Evans, J. Yamagishi, B. O'Brien *et al.*, "The voiceprivacy 2020 challenge: Results and findings," *Computer Speech & Language*, vol. 74, p. 101362, 2022.

[10] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[11] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.

[12] Z. Tian, L. Cui, J. Liang, and S. Yu, "A comprehensive survey on poisoning attacks and countermeasures in machine learning," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–35, 2022.

[13] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[14] R. A. Khalil, E. Jones, M. I. Babar, T. Jan, M. H. Zafar, and T. Alhussain, "Speech emotion recognition using deep learning techniques: A review," *IEEE access*, vol. 7, pp. 117 327–117 345, 2019.

[15] A. Nautsch, A. Jiménez, A. Treiber, J. Kolberg, C. Jasserand, E. Kindt, H. Delgado, M. Todisco, M. A. Hmani, A. Mtibaa *et al.*, "Preserving privacy in speaker and speech characterisation," *Computer Speech & Language*, vol. 58, pp. 441–480, 2019.

[16] B. M. L. Srivastava, N. Tomashenko, X. Wang, E. Vincent, J. Yamagishi, M. Maouche, A. Bellet, and M. Tommasi, "Design choices for x-vector based speaker anonymization," *arXiv preprint arXiv:2005.08601*, 2020.

[17] B. M. L. Srivastava, M. Maouche, M. Sahidullah, E. Vincent, A. Bellet, M. Tommasi, N. Tomashenko, X. Wang, and J. Yamagishi, "Privacy and utility of x-vector based speaker anonymization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 2383–2395, 2022.

[18] P. Champion, D. Jouvet, and A. Larcher, "A study of f0 modification for x-vector based speech pseudonymization across gender," *arXiv preprint arXiv:2101.08478*, 2021.

[19] I.-C. Yoo, K. Lee, S. Leem, H. Oh, B. Ko, and D. Yook, "Speaker anonymization for personal information protection using voice conversion techniques," *IEEE Access*, vol. 8, pp. 198 637–198 645, 2020.

[20] C. O. Mawalim, K. Galajit, J. Karnjana, and M. Unoki, "X-vector singular value modification and statistical-based decomposition with ensemble regression modeling for speaker anonymization system." in *Interspeech*, 2020, pp. 1703–1707.

[21] S. Meyer, P. Tilli, P. Denisov, F. Lux, J. Koch, and N. T. Vu, "Anonymizing speech with generative adversarial networks to preserve speaker privacy," in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 912–919.

[22] P. Champion, T. Thebaud, G. Le Lan, A. Larcher, and D. Jouvet, "On the invertibility of a voice privacy system using embedding alignment," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 191–197.

[23] N. Tomashenko, B. M. L. Srivastava, X. Wang, E. Vincent, A. Nautsch, J. Yamagishi, N. Evans, J. Patino, J.-F. Bonastre, P.-G. Noé *et al.*, "Introducing the voiceprivacy initiative," *arXiv preprint arXiv:2005.01387*, 2020.

[24] P. Champion, D. Jouvet, and A. Larcher, "Evaluating x-vector-based speaker anonymization under white-box assessment," in *Speech and Computer: 23rd International Conference, SPECOM 2021, St. Petersburg, Russia, September 27–30, 2021, Proceedings 23*. Springer, 2021, pp. 100–111.

[25] G. P. Prajapati, D. K. Singh, P. P. Amin, and H. A. Patil, "Voice privacy through x-vector and cyclegan-based anonymization." in *Interspeech*, 2021, pp. 1684–1688.

[26] Z. Cai, H. L. Xinyuan, A. Garg, L. P. García-Perera, K. Duh, S. Khudanpur, N. Andrews, and M. Wiesner, "Privacy versus emotion preservation trade-offs in emotion-preserving speaker anonymization," in *2024 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2024, pp. 409–414.

[27] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fr\'echet audio distance: A metric for evaluating music enhancement algorithms," *arXiv preprint arXiv:1812.08466*, 2018.

[28] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "Libritts: A corpus derived from librispeech for text-to-speech," *arXiv preprint arXiv:1904.02882*, 2019.

[29] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification." in *Interspeech*, vol. 2017, 2017, pp. 999–1003.

[30] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[31] H. Ruwali, "Skydocs/speaker-identification: Speaker identification using neural net." 2020. [Online]. Available: https://github.com/SkyDocs/speaker-identification

[32] A. Radford, J. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision (arxiv: 2212.04356). arxiv," 2022.

[33] K. Warren, T. Tucker, A. Crowder, D. Olszewski, A. Lu, C. Fedele, M. Pasternak, S. Layton, K. Butler, C. Gates *et al.*, ""' better be computer or i'm dumb": A large-scale evaluation of humans as audio deepfake detectors," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 2696–2710.

[34] A. Hamza, A. R. R. Javed, F. Iqbal, N. Kryvinska, A. S. Almadhor, Z. Jalil, and R. Borghol, "Deepfake audio detection via mfcc features using machine learning," *IEEE Access*, vol. 10, pp. 134 018–134 028, 2022.

[35] X. Yao and S. An, "Dp-voicepub: Differential privacy-based voice publication," in *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2023, pp. 1–5.