Results

Explanation of Metrics:

* Equal Error Rate(EER). the EER corresponds to the threshold θEER at which the two detection error rates are equal
* Log-likelihood-ratio cost function estimated by optimal calibration using monotonic transformation of the scores to their empirical LLR values.
* Word Error Rate(WER)
* OA: Ignorant Attacker: original enrollment and anonymized trial data are used for evaluation. We refer to this scenario as (original, anonymized) or oa in short. Users anonymize their trial data, but attackers are unaware of it, hence they use original data for enrollment.
* AA: Lazy-informed : anonymized enrollment and anonymized trial data are used for evaluation. We refer to this scenario as (anonymized, anonymized) or aa in short. This scenario reflects the situation when the enrollment data are anonymized data produced by users, who are assumed to use the same anonymization system but different pseudo-speakers from their trial data.5 While it is unlikely that attackers have access to anonymized data with explicit speaker identities, they may infer the identities of a subset of the data from the spoken contents and subsequently use this data as enrollment data. This scenario also reflects the alternative situation when attackers have access to original enrollment data and anonymize them us- ing the same system (which is assumed to be publicly available) so that they become more similar to the anonymized trial data. Here again, the data is anonymized using a different pseudo-speaker, since attackers do not know which pseudo-speaker was picked by each user. Hence, both situations result in the same attack model.
* AA(semi-informed): attack- ers have the same knowledge as in the previous case (the anonymization system, but not the pseudo-speaker picked by each speaker) and, in addi- tion to this, they anonymize the training set for the ASVeval model using the same anonymization system with different pseudo-speakers and re- train it on this data.
* LibriSpeech, VCTK: datasets used

|  |  |  |
| --- | --- | --- |
| Privacy Metric [Highest EER%] | Model | EER% |
| OA | F0-Modification | 53.37% |
| AA | GMM-PCA | 39.79% |
| OA Female | F0-Modification | ~52% |
| OA Male | A2 | ~58% |
| AA Female | GMM-PCA ,AAN2 | ~35% |
| AA Male | GMM-PCA | ~42% |

|  |  |  |
| --- | --- | --- |
| Highest | Model |  |
| OA | A1,X-vector-Pool, F0-Modification | ~1.0 |
| AA | GMM-PCA, AAN2 | 0.8~1.0 |

|  |  |  |
| --- | --- | --- |
| Speech Recognition Error  [Lowest WER%] | Model | WER% |
| on VCTK-Test | F0-Modification | 14.6% (Original 12.8%) |
| on LibriSpeech-Test | Formant-Shifting | 5.8% (Original 4.1%) |

|  |  |  |
| --- | --- | --- |
| Best Performance Against Attack Model [Highest EER%] | Model | EER% |
| OA | F0-Modification | 53.23% |
| AA | GMM-PCA | 39.34% |
| AA(semi-informed) | F0-Modification | 22.62% |

|  |  |  |
| --- | --- | --- |
| Utility Metric [Lowest WER%] Test Set | Model | WER% |
| LibriSpeech-Test O | X-vector-Pool | 6.73% |
| LibriSpeech-Test A | Signal-Processing | 4.34% |
| VCTK-Test O | X-vector-Pool, ANN | 15.23% |
| VCTK-Test A | Voiceprint-Perturbation | 9.09% |

1. Voice Distinguishable Metric:

Models that have Distinct Diagonal: Voiceprint-Perturbation, Formant-Shifting, B2

1. Gain of Voice Distinctiveness against De-identification performance:

Voiceprint-Perturbation

1. EER% against WER(%) low WER% with High EER%: Model ANN

|  |  |
| --- | --- |
| Metric | Model |
| Highest Naturalness | Formant-Shifting |
| Highest Intelligibility | Formant-Shifting |
| Lowest Similarity Same Speaker | ANN |
| Lowest Similarity Different Speaker | Voiceprint-Perturbation |

**Note: Except Model Formant Shifting, Voice Print Perturbation and Signal Processing, all models are based on x-vector (identity vector)**

**Model Formant-Shifting Pseudonymisation Method**

Proposes to use a deterministic and reversible pseudonymization method that uses signal processing based on formant-shifting to hide speaker identity

Consists of 3 steps sequentially

1. Simulate a different vocal tract for the speaker
   1. Formant frequency locations and amplitudes are modified to match those of a desired speaker

* This also allows modification of the amplitude of the formant through thte filter’s gain.
* The median of the parameters VTL, formant frequencies and their amplitudes are pre-computed per speaker by aggregating across several of their utterance.
  + - First step: the playback speed of the audio is altered so that the formant frequencies are shifted by a linear factor - Simulate a different vocal tract length (VTL)
    - Second step: the individual formant frequencies are further shifted to the desired vales as follows for each formant – creates a version of the source signal with formants at the desired locations
      1. Centre two band-pass Han filters, one at the current formant location and the other at the desired location
      2. Extract and swap the spectral contents of the two locations

1. Change the speaking rate and fundamental frequency
   1. The speaking rate is estimated by an existing method that automatically locates syllables from speech without using transcription. The method uses peaks in the signal energy, as cues for syllables
      * Reference: N. H. De Jong and T. Wempe, “Praat script to detect syllable nuclei and measure speech rate automatically,” Behavior research methods, vol. 41, no. 2, pp. 385–390, 2009.
   2. The fundamental frequency and the speaking rate are changed by using a pitch synchronous overlap and add method
      * Reference: E. Moulines and F. Charpentier, “Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones,” Speech communication, vol. 9, no. 5-6, pp. 453–467, 1990.
2. Additional Processing to hide the speaker identify
   1. Exchange the F4 and F5 bands by using Hann filter method
      1. Centre two band-pass Han filters, one at the current formant location and the other at the desired location
      2. Extract and swap the spectral contents of the two locations
   2. Add modulated pink noise at the speaker’s F6-F9 bands to mask these formants

**Model Voiceprint-Perturbation**

System is based on Voice-Indistinguishability, voiceprint perturbation mechanism and privacy-preserving speech synthesis framework

1. Voice Indistinguishability
   1. Rigorous privacy metric for voiceprint privacy
   2. use x-vector for representation of voiceprint
   3. Different x-vector can be seen as different speaker identity
   4. Apply voice indistinguishability on the pool of x vector databases
   5. Guarantees that given the output x-vector database D’, an attacker hardly distinguishes whether the original x-vector database is D or D’
2. Voiceprint Perturbation Mechanism
   1. X-vector database construction
      1. Given a speech database, for each speaker with numerous utterances, use one extracted x-vector, which is the mean of the x-vectors extracted from these utterances, use it to represent this speaker’s identity permanently
      2. Obtain an x-vector database D where each x-vector refers to one speaker
   2. Perturbation
      1. Given an input x-vector, the mechanism K perturbs x by randomly selecting an x-vector x’ in the dataset D according to calibrated probability distributions, providing plausible deniability for x
      2. Mechanism K: associated equation refers to paper
3. Privacy-preserving Speech synthesis framework

Diagram

Description automatically generated

* 1. After obtaining the perturbed x-vector database, synthesize the perturbed x-vector and original speech characteristics without the original x-vector
  2. For utterances of one speaker, use the same x-vector given by perturbed x-vector database
  3. Use a privacy-preserving speech synthesis framework to synthesis the perturbed x-vector and the original speech characteristic other than x-vector

Framework uses two modules to generate the speech data

* + - 1. An End-to-End acoustic model that produces a Mel-spectrogram
      2. A waveform vocoder based on Griffin-Lim algorithm that produces a speech waveform given the Mel-spectrogram after converting Mel-spectrogram to linear scale spectrogram using an inverse matrix

**Model Signal-Processing**

**Diagram

Description automatically generated**

It employs the McAdams’ coefficient to achieve anonymization

* Shifting the pole positions derived from the linear predictive coding (LPC) analysis of speech signals.

1. Starts with the application of frame-by-frame LPC source-filter analysis to derive LPC coefficients and residuals.
2. The residuals are set aside for later resynthesis, whereas LPC coefficients are converted into pole positions by polynomial root-finding.
3. The McAdams’ transformation is then applied to the angles of the poles (with respect to the origin in the z- plane), each one of which corresponds to a peak in the spectrum
4. While real-valued poles are left unmodified, the angles φ of the poles with a non-zero imaginary part (with values between 0 and π radians) are raised to the power of the McAdams’ coefficient α so that the transformed pole has new, shifted angle φα.
5. The value of α implies a contraction or expansion of the pole positions around φ = 1.
6. Corresponding complex conjugate poles are similarly shifted in the opposite direction and the new set of poles, including original real-valued poles, are then converted back to LPC coefficients.
7. Finally, LPC coefficients and residuals are used to resynthesize a new speech frame in the time domain.
8. This technique shares some similarities with the frequency warping-based methods. Except it modifies only the spectral envelope (not the pitch)

**Model X-vector-Pool**

Diagram

Description automatically generated

It comprises three steps:

(1) x- vector, pitch (F0) and bottleneck (BN) feature extraction

(2) x-vector anonymization

(3) speech synthesis (SS) using the anonymized x-vector and the

original F0 and BN features.

Step (1) encodes the spoken content by 256- dimensional BN features extracted using a TDNN-F ASR AM trained on the LibriSpeech train-clean-100 and train-other-500 datasets and speaker information by a 512-dimensional x-vector extracted using a TDNN

Step (2) computes an anonymized x-vector for every original x-vector. It is generated by averaging a set of N∗ x-vectors selected at random from a larger set of N x-vectors, itself composed of the N farthest x-vectors in the LibriTTS train- other-500 dataset, according to PLDA distance.

Step (3) uses a SS AM to generate Mel-filterbank features from the anonymized x-vector and the original F0 and BN features, and a neural source-filter (NSF) waveform to synthesize a speech signal from the anonymized x-vector and the F0 and Mel-filterbank features. The SS AM and NSF models are both trained on the LibriTTS train-clean-100 dataset.

**Diagram

Description automatically generatedModel Singular-Regression**

A1: Modifying the singular value of the input x-vectorDiagram

Description automatically generated

1. X-vector Pool Construction: constructed the x-vector pool to obtain the input x-vector and pseudo target x-vector The pseudo target x-vector was determined from the least similar centroid using a clustering method
2. Matrix Formation: An x-vector matrix was constructed using the x-vector of all available utterances of a speaker. The output is the x-vector matrix for the pseudo target x-vectors with dimension M\*N, M is the number of the utterances and N is the dimension of x-vector
3. Singular Value Decomposition and Modification: decomposition formula refer to paper
4. X-vector Reconstruction

A2: Decomposing the input x-vector based on its statistical properties and transforming it with regression models

Timeline

Description automatically generated

1. X-vector Variant-based Decomposition: low-variant x-vector is a stable part of the x-vector that contains the uniqueness of the speaker identity
2. Anonymization pool construction: build clustering models to create pseudo-target x-vectors, pseudo-target x-vector was determined by the centroid least similar to the pseudo-input x-vector. The pseudo-target x-vectors were fit into a regression model in two consecutive processes and then pairs of pseudo input-target x-vectors were fir into the regression model. Define the x-vector pairs as anonymization pool
3. Ensemble Learning for Regression Modelling: two ensemble regression models were constructed. Fit a linear function for transforming the original low-variant x-vector to the anonymized low-variant x-vector
4. Anonymized X-vector Reconstruction: concatenated the high-variant and low-variant anonymized x-vectors to from the anonymized x-vector

**Model F0-Modification:**

Alter the F0 paralinguistic information in an vector based speech pseudonymization system

Diagram

Description automatically generated

Model based on anonymizing speech utterances using x-vectors and neural waveform models

Idea: Separate speaker identity and linguistic content from an input speech utterance

Obtain x vector that encode the speaker identity

Module A: Feature extraction

Module B: Anonymization

* Derive a new pseudo-speaker identity using knowledge gleaned from a pool of external speakers

Module C: Speech synthesis

* Synthesizes a speech waveform from the pseudo-speaker x-vector together with the original features

Module D: Modification of F0 values

* Manipulating the F0 vales will impact both the speech synthesis acoustic model and vocoder models to transform the speech signal
* Modify the F0 values of the source uerance from a given speaker by using linear transformation
* : log-scaled F0 of the source speaker at the fram t
* : mean for the source speaker
* : standard deviation for the source speaker
* : mean for the log-scaled F0 for the pseudo-speaker
* : standard deviation for the log-scaled F0 for the pseudo-speaker

**Model GMM-PCA**

Diagram

Description automatically generated

1. Input audio
2. Derives three components from the audio
   1. X-vector extracted by TDNN (Time delay neural network)
   2. The bottleneck features obtained by applying an automatic speech recognition acoustic model
   3. Pitch information (F0)
3. The pseudo x-vector represents the new identity of the speaker and is used for all utterances intended to be spoke by that identity
4. A speech synthesis module use F0 an BN features and the pseudo-x-vector to generate melspectrograms, outputs mel-filterbanks
5. NSF model processes these filterbanks, along with the F0 and the pseudo x-vector, generating the anonymized audio

Problem: in the generation process of the pseudo x-vector, averaging of several x-vectors reduce diversity of voices and lead to the reduction in entropy

x-vector generation

1. Learn the properties of x-vector space suing PCA – Principle Component Analysis on a large x-vector dataset
   1. PCA maps original data matrix onto an orthogonal space
   2. Let us extract a set of factors and then select a subset of those factors, reduce the dimensionality of the problem, those factors are the principle components (PCs) extracted through PCA)
2. Fit a generative model on the PCA-reduced space, in order to sample from it

Sample from the GMM(Gaussian Mixture Model) in the PCA reduced space and then apply the PCA inverse transform

* GMM: a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distribution with unknown parameters

Result:

Table

Description automatically generated

(O-O) : baseline

(O-A): Original enrollment and anonymized trail

* Examines the difference between the original voices and an anonymized version of them
* EER decreasing up to 6.69%

(A-A): anonymized enrollment and anonymized trail

* Each of the enrollment and trail utterances are anonymized but to different identities

Improving the diversity of anonymize voices

C approaches a perfect score in many scenario, indicating strong anonymization

WER results 0.36% to 3.63%

**Model AAN: Autoencoder-Adversarial Network**

Diagram

Description automatically generated

Speaker de-identification method using DAT and Autoencoders

3 parts

* + 1. Feature Extraction
    2. X-vector anonymization
    3. Speech Synthesis

Speaker characteristics-invariant approach based on Autoencoder Adversarial Network (AAN)

Encoder-decoder autoencoder branch tries to reconstruct he input x-vector wile in adversarial branches try to mitigate speaker characteristics, such as gender, accent and speaker identity

Hide the speaker identity when reconstructing the x-vector by means f te autoencoder

Making the encoded representation ivariant to the domain f speaker characteristics by using a Domain Adversarial neural Network (DANN)

AAN1: AAN used as x-vector anonymizer

* The x-vectors extracted from the baseline are used as input to the autoencoder that generates as output the pseudo-speaker x-vector

AAN2: Transform the pseudo-speaker x-vector generated by the baseline

* The anonymized x-vector is used as input to the autoencoder that generates as output a new anonymized x-vector

**Similar Model to this ANN:** **CycleVAE-GAN**

* Diagram

  Description automatically generatedz is the latent vector which corresponds to the linguistic information of the input speech
* x is the input speech given by source speaker
* x’ is the reconstructed speech given I­X is the source speaker identity vector/is the converted speech given I­X being the target speaker (here means the anonymous speaker) identity vector
* x’’ is the converted back speech which should recover the original input speech x
* dashed line represent the cyclic conversion path that produces x’’
* When the speaker identity vector being replaced with an anonymous speaker identity vector, the speech can be anonymized

By minimizing the loss function of VAE (refer to [15] for the formula),

* the encoder is trained to extract latent vector z (contains the linguistic information of the input speech)
* the decoder is trained to reconstruct the input speech from the latent vector z and the source speaker identity I­X

The target speaker identity vector: a one-hot vector containing 1 for the target speaker and 0s for other speakers, which is fed into the decoding process of the VAE to convert the source speaker speech to the target speaker speech

Optimize the VAE module by minimizing the loss function of CycleAVE (refer to [15] for the formula) which is a combination of the VAE loss function and the cycle-consistency loss function

GAN part:

The GAN module is used to train the CycleVAE-GAN model

The decoder of the VAE is considered as the generator of the GAN

The discriminator of GAN helps the generator( VAE decoder\_ to produce a speech similar to that of the target speaker

Anonymous Speaker Identity Vector

a) One-hot vector: used for converting to target speaker

To minimize the voice biometric of a given speaker

Table

Description automatically generatedUse uniform values

* 1. As in b) assign 0 to the source speaker and assign the value of 1/(n-1) to other speakers uniformly in the anonymous speaker identity vector
  2. As in c) assign -1 to the source speaker and assign the value of 2/(n-1) to other speakers uniformly in the anonymous speaker identity vector
  3. As in d) assign -1/(n-2) to the source speaker and 1/(n-2) to the other speaker
  4. All leads to summation of 1: the decoder of the VAE is trained to handle a speaker identity vector is a unit vector

Use nonuniform values

Table

Description automatically generated

1. As in a) use the cosine similarity of each pair of vectors
2. As in b) use the inverse of the cosine similarity values, setting the source speaker value to 0 and normalizing the values to obtain a sum of 1
3. Determination of k-farthest speakers and uniformly assigning the value of 1/k to them and 0s to others:
   1. C) one-farthest speaker is set to 1 and others are set to 0
   2. D) two-farthest speaker is set to 0.5 each and other are set to 0

Result

* Dataset: VCC2016: 10 speakers (5male, 5female) with 162 and 54 utterance for training and testing for each speaker
* Models: Speaker Anonymization (CycleVAE-GAN) + speech synthesis (WORLD vocoder)

Baseline Result

1. Model: Gaussian Mixture Model (GMM)
2. Model: Deep Neural Network (DNN)

* **Dataset**: VCC 2016 test set

**Metric**: Speaker Identification Accuracy | GMM: 100%, DNN: 100%

**Conclusion**: Both systems were trained well

* **Dataset**: Voice converted version of VCC 2016 test data: each utterance from a speaker in the rest set was converted to rest nine speakers’ voice using CycleVAE-GAN

**Metric**: Speaker Identification Accuracy | GMM: 99.8%, DNN: 94.1%

**Conclusion**: CycleVAE-GAN can successfully modify the identify fo a source speaker to that of the target speaker

Table

Description automatically generated



Reduced the identification accuracy of speech data to 0.1%-9.2%

While retain the speech recognition accuracy to approximately 70%15]

Did not mention time

s

References

Model Singular Regression:

Mawalim, C. O., Galajit, K., Karnjana, J., & Unoki, M. (2020). X-vector

singular value modi cation and statistical-based decomposition with ensemble regression modeling for speaker anonymization system. In Interspeech (pp.1703{1707).

Model Formant-Shifting:

Dubagunta, S. P., van Son, R. J., & Doss, M. M. (2020). Adjustable deterministic

pseudonymisation of speech: Idiap-NKI's submission to VoicePrivacy 2020 challenge. https://www.voiceprivacychallenge.org/docs/Idiap-NKI.pdf.

Model Voiceprint-Perturbation:

Han, Y., Li, S., Cao, Y., & Yoshikawa, M. (2020b). System description for VoicePrivacy Challenge. Kyoto team. https://www.voiceprivacychallenge.org/docs/Kyoto.pdf.

Model F0-Modification:

Champion, P., Jouvet, D., & Larcher, A. (2020b). A study of F0 modi cationfor x-vector based speech pseudo-anonymization across gender. https://hal.archives-ouvertes.fr/hal-02995862.

Model GMM-PCA:

Turner, H., Lovisotto, G., & Martinovic, I. (2020). Speaker anonymization with distribution-preserving x-vector generation for the VoicePrivacy Challenge 2020. arXiv preprint arXiv:2010.13457 , .

Model ANN:

Espinoza-Cuadros, F. M., Perero-Codosero, J. M., Ant on-Mart  n, J., & Hern andez-G omez, L. A. (2020a). Speaker de-identi cation system using autoencoders and adversarial training. arXiv preprint arXiv:2011.04696 , .

Results: The VoicePrivacy 2020 Challenge: Results and Findings