Voice Conversion Attacks on Speaker De-Identification Schemes

Shufei He and Peter Roelse
Irdeto B.V.
The Netherlands
Email: {shufei.he,peter.roelse}@irdeto.com

Abstract—Speaker de-identification can be used to protect the privacy of a speaker. One class of de-identification schemes uses voice conversion techniques to transform utterances of a source speaker into utterances of a target speaker. This paper presents two new generic attacks on such schemes. The attacks have been implemented for Gaussian mixture model-based schemes. Experimental results show that the first new attack achieves success rates that are considerably higher than the rates of existing attacks. The second new attack shows that these rates can be increased further if the attacker has access to an utterance of the target speaker.

Index Terms—Security, privacy, speaker de-identification.

I. INTRODUCTION

Speaker de-identification aims to remove speaker-dependent voice features from spoken utterances so that individual speech data cannot be linked with specific individuals [1]. This technique can therefore be used to minimize the privacy risk associated with creating, using, archiving, sharing, and publishing speech data. Speaker de-identification is important for individuals who are concerned about their privacy and for organizations that wish or need to process speech data while preserving speaker privacy. For example, speaker de-identification can provide anonymity protection for individuals in their publicly released voice recordings (e.g., for witnesses, whistle blowers, or journalist’s sources) and in their interactions with voice controlled personal assistants (such as Siri and Google Assistant). Additionally, speaker de-identification enables processing of speech data previously prohibited by data protection laws and regulations [2]–[4].

Existing speaker de-identification schemes can be divided into two different types: one type of scheme is based on voice conversion techniques [5] and the other type is based on speech recognition techniques (such as speech-to-text and speech-to-phoneme) [6]. An advantage of the latter type of scheme is that it achieves a perfect de-identification in that all the speaker’s voice features are removed; however, a drawback is that the synthetic speech contains a certain amount of information loss due to recognition errors. For example, Microsoft’s speech recognition system yields a 5.1% word error rate on the 2000 Switch-board evaluation set [7]. This paper focusses on de-identification schemes based on voice conversion. Voice conversion is a technique that modifies speech produced by a source speaker so that it is perceived by listeners as if it had been uttered by a different speaker, referred to as the target speaker. Currently, the Gaussian Mixture Model (GMM)-based scheme as introduced in [8] is the most popular scheme for voice conversion (see e.g. [5], [9], [10]).

In the setting of this paper, the attacker is given a set of possible source speakers, an utterance of each of the speakers in the set, and a de-identified utterance of one of the speakers in the set. The objective of the attack is to re-identify this speaker. Existing re-identification attacks apply an automated speaker identification system to the de-identified utterance and the utterances of the source speakers to determine the most likely speaker from the set of source speakers (see e.g. [5], [9], [10]). Re-identification attacks are also used to evaluate the effectiveness of speaker de-identification schemes. To this end, the attack is executed a number of times, each time with a different de-identified utterance of a possible source speaker as input, and its effectiveness is then defined as the fraction of executions in which the speaker was re-identified correctly. This fraction is the success rate of the attack and is also referred to as the re-identification rate.

This paper presents two new generic re-identification attacks on speaker de-identification schemes. Both attacks are based on voice conversion, and one of the attacks assumes a stronger attack model in which an utterance of the target speaker is known. The attacks have been implemented using a GMM-based voice conversion method and a GMM-based identification scheme, and experiments were conducted using the LibriSpeech dataset [11]. In particular, it is shown that the first new attack can achieve a re-identification rate of 19% if the source speaker set contains 120 speakers. This is considerably higher than the rate of existing attacks that directly apply an identification system to the de-identified utterance, which equals around 4% in the experiments. The second new attack exploits the availability of an utterance of the target speaker to increase this rate to 26% for the set of 120 speakers. Furthermore, if the designer uses a publicly available text-to-speech voice as the target speaker and if the attacker recognizes this voice, then the second new attack can use the speech-to-text method to achieve a rate of 86% for this set of speakers.

This paper is organized as follows: Section II describes the basic concepts of GMM-based speaker identification and GMM-based voice conversion. The attacks are described in Section III and the experimental results are presented in Section IV. Finally, conclusions can be found in Section V.
A. Speaker Identification

Throughout the paper, the set of source speakers is denoted by \( S = \{1, 2, \ldots, N\} \). Moreover, an utterance is a segment of speech of one speaker, and an utterance may be obtained by concatenating a number of shorter utterances of the same speaker. The description of the attacks and the experiments use a GMM as the speaker model for speaker identification. A Gaussian mixture density function \( p : \mathbb{R}^n \rightarrow \mathbb{R} \) is defined as a weighted sum of \( m \) Gaussian density functions \( g_i : \mathbb{R}^n \rightarrow \mathbb{R} \) with \( i = 1, 2, \ldots, m \). If the weights are denoted by \( w_i \in \mathbb{R} \) with \( 0 < w_i < 1 \) and \( \sum_{i=1}^{m} w_i = 1 \), and if \( \mu_i \in \mathbb{R}^n \) and \( \Sigma_i \in \mathbb{R}^{n \times n} \) denote the mean vector and the covariance matrix of \( g_i \) respectively, then the GMM is characterized by the set \( \{(w_i, \mu_i, \Sigma_i)|1 \leq i \leq m\} \). This set is denoted by \( \lambda \) in the following text and is also referred to as a speaker model. Using this notation,

\[
p(x|\lambda) = \sum_{i=1}^{m} w_i g_i(x).
\]

The speaker identification process comprises two phases: a training phase and a testing phase. The training phase is a one-time setup phase per source speaker. During this phase, a set of feature vectors is extracted from the training utterance of the source speaker. The experiments in this paper use the Mel-Frequency Cepstral Coefficients (MFCCs) extraction method [12] for this purpose. Inputs to this method are the training utterance of the source speaker. The experiments in this paper outputs an arbitrary element (i.e. speaker) of the set \( \arg \max_{j \in S} \sum_{t=1}^{T} \log p(x_t|\lambda_j) \).

For detailed information, the reader is referred to [13].

B. Voice Conversion

Speaker de-identification and the new attacks are both based on voice conversion, transforming utterances of a source speaker into utterances of a target speaker. The GMM-based method for voice conversion as presented in [15] is used in the experiments in this paper. The voice conversion process is also divided into a training phase and a testing phase. The training phase determines a conversion function that maps acoustic characteristics of the source speaker to characteristics of the target speaker. To this end, first two sets of feature vectors are extracted: one from the training utterance of the source speaker and one from the training utterance of the target speaker. Line Spectral Frequencies (LSFs) (see e.g. [16]) are used as feature vector representations in the experiments in this paper. Next, the frames of the source and target utterances are aligned. Let \( x_t \in \mathbb{R}^d \) and \( y_t \in \mathbb{R}^d \) denote the feature vectors of the harmonic component at Frame \( t \) for \( 1 \leq t \leq K \) of the source speaker and the target speaker respectively, and let \( z_t \in \mathbb{R}^{2d} \) be defined as the concatenation of \( x_t \) and \( y_t \). The distribution of the vectors \( z_t \) for \( 1 \leq t \leq K \) is modelled by a Gaussian mixture density function with \( k \) Gaussian density functions \( g_i : \mathbb{R}^{2d} \rightarrow \mathbb{R} \) (see also (1)):

\[
p(z|\lambda) = \sum_{i=1}^{k} w_i g_i(z).
\]

The mean vector \( \mu_i \) and the covariance matrix \( \Sigma_i \) of \( g_i \) are written as:

\[
\mu_i = \begin{pmatrix} \mu_i(x) \\ \mu_i(y) \end{pmatrix} \quad \text{and} \quad \Sigma_i = \begin{pmatrix} \Sigma_i^{(xx)} & \Sigma_i^{(xy)} \\ \Sigma_i^{(yx)} & \Sigma_i^{(yy)} \end{pmatrix},
\]

with \( \mu_i(x), \mu_i(y) \in \mathbb{R}^d \) and \( \Sigma_i^{(xx)}, \Sigma_i^{(xy)}, \Sigma_i^{(yx)}, \Sigma_i^{(yy)} \in \mathbb{R}^{d \times d} \). As in the speaker identification process, the parameters are estimated using the EM algorithm. The initial values for this algorithm are computed as in [16]. The conversion function \( F : \mathbb{R}^d \rightarrow \mathbb{R}^d \) is then defined as:

\[
F(x) = \sum_{i=1}^{k} p_i(x) \left( \mu_i(y) + \Sigma_i^{(yx)} (\Sigma_i^{(xx)})^{-1} (x - \mu_i(x)) \right),
\]

where \( p_i(x) \) is the probability of \( x \) belonging to the \( i \)-th Gaussian component of the GMM associated with \( x \).
speaker. These feature vectors are then input to a speech synthesis procedure. The above description applies to the harmonic component of the frames; for a complete description of the voice conversion method used in the experiments, including the stochastic component and pitch level conversion, refer to [16].

Training data are called “parallel” if the contents of the utterance of the source speaker and the utterance of the target speaker are the same. If this is the case, then the accuracy of the voice conversion will generally be improved compared with non-parallel training data [16]. The experiments in this paper therefore use parallel data in the training phase of the de-identification process. Observe that it is generally convenient to use a text-to-speech voice (see e.g. [17]) as the target speaker in a practical deployment of a de-identification scheme since this enables the construction of parallel training data from the utterances of the source speakers.

The de-identification process in this paper adopts the N-to-one conversion model in which each source speaker in $S$ is mapped to the same target speaker. Instead, one could use a one-to-one conversion model in which each source speaker is mapped to a different target speaker. However, a drawback of this model is that it is likely to be easier for the adversary to determine if different de-identified utterances are associated with the same source speaker. This model is therefore not considered in this paper.

The experiments assume that the target speaker is not an element of $S$. A reason for excluding the target speaker from the set is that an attacker could first identify this speaker in the set, using the property that de-identified utterances are similar to an utterance of the target speaker. Next, the attacker can exclude the target speaker from the set. In addition, as shown in the next section, after the target speaker has been identified, the attacker could use an utterance of the target speaker to attack the scheme.

III. RE-IDENTIFICATION ATTACKS

As before, the set of source speakers is denoted by $S = \{1, 2, \ldots, N\}$. Further, all de-identified utterances are assumed to be generated using the voice conversion method as described in Section II-B. The attacker is assumed to have access to $S$ and an utterance of each of the speakers in $S$. For example, the attacker may have compromised the de-identification system to obtain such information. Alternatively, the attacker may select the source speakers in the set and obtain an utterance from each of the speakers in some other way. The objective of the attacker is: given $S$ with corresponding utterances and a de-identified utterance of $i \in S$, determine $i$.

The following text presents three attacks. The description of the attacks assumes that the speaker identification scheme of Section II-A is used. Attack 1 takes the de-identified utterance and the set of possible source speakers $S$ with corresponding utterances as inputs and directly applies this speaker identification scheme to determine a speaker with the highest similarity score. Attack 1 is similar to the evaluation methods as used in [5], [9], [10].

Attack 1

Inputs:
- The source speaker set $S = \{1, 2, \ldots, N\}$ and an utterance of each $j \in S$
- A de-identified utterance of $i \in S$

Output: A speaker $k \in S$

1. Extract a set of feature vectors $\{x_1, x_2, \ldots, x_T\}$ for identification from the de-identified utterance
2. For $j = 1, 2, \ldots, N$
   a. Use the utterance of $j$ to train speaker model $\lambda_j$
   b. Compute $P_j = \sum_{t=1}^{T} \log p(x_t|\lambda_j)$
3. Select an arbitrary element $k$ of the set $\arg \max_{j \in S} P_j$
4. Output $k$

Note that an execution of Attack 1 is successful if and only if $k = i$. Attack 2 is the first new attack. This attack assumes that the attacker chooses a voice conversion method. For a source speaker in the set, the attack then uses this method to train a conversion function that converts the de-identified utterance into an utterance of the source speaker. Next, a similarity score between the resulting converted utterance and the original (i.e. non-converted) utterance of the speaker is computed. The attack repeats these steps for every speaker in $S$ and outputs a speaker with the highest score. The intuition behind this attack is that a Speaker $i$ – target speaker – Speaker $j$ converted utterance will be most similar to a Speaker $j$ utterance if $i = j$.

Attack 2

Inputs:
- The source speaker set $S = \{1, 2, \ldots, N\}$ and an utterance of each $j \in S$
- A de-identified utterance of $i \in S$
- A voice conversion method

Output: A speaker $k \in S$

1. For $j = 1, 2, \ldots, N$
   a. Use the de-identified utterance, the utterance of $j$, and the voice conversion method to determine a voice conversion function
   b. Use this function to convert the de-identified utterance into a new utterance of $j$
   c. Extract a set of feature vectors $\{x_1, x_2, \ldots, x_T\}$ for identification from this new utterance
   d. Use the utterance of $j$ to train speaker model $\lambda_j$
   e. Compute $P_j = \sum_{t=1}^{T} \log p(x_t|\lambda_j)$
2. Select an arbitrary element $k$ of the set $\arg \max_{j \in S} P_j$
3. Output $k$

The third attack uses a stronger attack model than the first two attacks: it assumes that the attacker also has access to an utterance of the target speaker. For example, the attacker
may have obtained such an utterance by compromising the system implementing the speaker de-identification scheme. Alternatively, the attacker may recognize the target speaker from a de-identified utterance and obtain a target speaker utterance in some other way. The access to the utterance of the target speaker and the use of a voice conversion method enable the attacker to perform the following voice conversion attack: for a speaker in the source speaker set, train a voice conversion function that converts the utterance of this speaker into an utterance of the target speaker. Then use this converted utterance to train a “virtual speaker” model and compute the similarity score with the de-identified utterance. The attack repeats this process for every speaker in \( S \) and outputs a speaker with the highest score.

**Attack 3**

**Inputs:**
- The source speaker set \( S = \{1, 2, \ldots, N\} \) and an utterance of each \( j \in S \)
- A de-identified utterance of \( i \in S \)
- A voice conversion method
- An utterance of the target speaker

**Output:** A speaker \( k \in S \)

1. Extract a set of feature vectors \( \{x_1, x_2, \ldots, x_T\} \) for identification from the de-identified utterance
2. For \( j = 1, 2, \ldots, N \)
   a. Use the utterance of \( j \), the utterance of the target speaker, and the voice conversion method to determine a voice conversion function
   b. Use this function to convert the utterance of \( j \) into an utterance of the target speaker
   c. Train a virtual speaker model \( \lambda_j^* \) using the converted utterance
   d. Compute \( P_j = \sum_{t=1}^{T} \log p(x_t|\lambda_j^*) \)
3. Select an arbitrary element \( k \) of the set \( \arg \max_{j \in S} P_j \)
4. Output \( k \)

In [18], Newton et al. proposed an attack on de-identified facial images, referred to as parrot recognition, that is similar to Attack 3. For details, refer to [18].

To measure the effectiveness of an attack, the attack is repeated with different de-identified utterances of source speakers in \( S \) as input. Next, the re-identification rate \( R \) is computed as:

\[
R = \frac{\text{successful executions}}{\text{#executions}}. \tag{2}
\]

The next section presents an experimental evaluation that determines re-identification rates of the three attacks. These re-identification rates will also be compared with the identification rate of a baseline experiment that identifies speakers based on utterances that have not been de-identified and with the expected rate of the attack based on random guessing, outputting a random element of the set \( S \). The expected rate of the attack based on random guessing equals \( 1/|S| \) and the rate of the baseline experiment is defined as in (2).

### IV. Experiments

#### A. Settings

1) **Source Speakers and Target Speaker:** The first 120 females from the LibriSpeech train-clean-100 dataset [11] are used to construct source speaker sets. The speech data in a single LibriSpeech file is associated with one speaker and is referred to as a “short utterance” in the following text. Recall from Section III that the first input of an attack contains an utterance of each of the source speakers. Each of these utterances, i.e. one utterance for each of the 120 source speakers, is chosen to be approximately 2 minutes long and is obtained by concatenating a suitable number of short utterances of the corresponding source speaker. These utterances are also input to the baseline experiment. The adult male from the Microsoft text-to-speech voices [17] is used as the target speaker.

2) **De-Identified Utterances:** The voice conversion method described in Section II-B is used to generate a de-identified utterance for each of the 120 source speakers. For each speaker, first a number of short utterances with a total duration of approximately 2 minutes are selected for training and a number of short utterances with a total duration of approximately 1 minute are selected for testing. There is no overlap between the short utterances used for training and the ones used for testing to evaluate the effectiveness of the de-identification scheme on unseen data. In addition, it is not assumed that the attacker has access to short utterances that were used for speaker de-identification. More precisely, there is no overlap between the short utterances used for de-identification and the short utterances of Section IV-A1. To optimize the accuracy of the de-identification process, parallel data are used for training. The LibriSpeech dataset does not provide such data but it does provide text transcriptions of all its speech data. These text transcriptions and the text-to-speech synthesis method of [17] are used to produce parallel training data of the target speaker. Next, the parallel data are used to train a voice conversion function for each source speaker, and this conversion function is then applied to the short test utterances of the speaker. Finally, the outputs of the conversion process are concatenated to obtain one de-identified utterance of around 1 minute for the speaker. These de-identified utterances are inputs to the attacks (i.e. one de-identified utterance per execution, used as the second input of each attack) and the corresponding original utterances are inputs to the baseline experiment.

In the voice conversion system, the sampling rate of the speech data equals 16 kHz, the number of samples per frame is 128, and a Hamming window is used for windowing. In addition, the pitch range is set from 60 to 500 Hz, the maximum voiced frequency is 5 kHz, and the Montreal forced aligner [19] is used for speech-text alignment. Moreover, as in [16], LSFs are extracted using a 14th order LPC filter and the number of Gaussian density functions, i.e. the value of \( k \) in Section II-B, is chosen to be 16. The EM algorithm uses 10 iterations to estimate the parameters of the conversion function.
3) Voice Conversion in Attacks 2 and 3: The third input of Attacks 2 and 3 is a voice conversion method. The experiments use the same voice conversion method for the attacks as used for de-identification. A motivation for this is that it is in the interest of both the designer and the attacker to choose the best-known method for voice conversion. Since there are only a few effective methods available, it is therefore not unlikely that they choose the same method and that the values of the parameters are comparable, e.g., because they both use values that have been reported in literature as being good choices.

In Attack 2, it is assumed that the attacker only has access to non-parallel data for training the voice conversion function. Attack 3 will be performed for both cases, i.e., for non-parallel and for parallel data. For example, the latter case applies if the designer uses a text-to-speech voice as the target speaker and if the attacker obtains access to this text-to-speech method. The availability of parallel data allows the attacker to also optimize the accuracy of the voice conversion process. Note that the description of Attack 3 in Section III does not assume that parallel data are available, and that this description may need to be adapted for cases in which parallel data are available.

If the attacker uses non-parallel training data, then the number of Gaussian density functions is chosen to be 8 (as in [16]). In addition, in this case the text-to-speech synthesis method of [17] is used to generate an utterance of the target speaker with a duration of approximately 1 minute that is the fourth input of Attack 3. This duration ensures that the attacker has approximately the same amount of (non-parallel) data for training the voice conversion function as in Attack 2. For the parallel variant of Attack 3, this text-to-speech method and the text transcriptions of the source speakers’ input utterances are used to generate parallel training data. As above, the number of Gaussian density functions for parallel data is chosen to be 16. The values of the other parameters are chosen to be equal to the ones used in the speaker de-identification process.

4) Speaker Identification: The experiments use the speaker identification scheme as described in Section II-A in the baseline experiment and in the attacks. The number of Gaussian density functions, i.e., the value of $m$ in Section II-A, equals 64 and the number of MFCCs, i.e., the value of $n$ in Section II-A, equals 39. The MFCCs include delta and delta-delta coefficients (refer to [12] for detailed information). Moreover, the sampling rate of the speech data is 16 kHz, the number of samples per frame equals 256, the overlap between adjacent frames is 50%, and a Hamming window is used for windowing. Finally, the EM algorithm uses 10 iterations to estimate the parameters of a speaker model.

B. Re-Identification Rates

The experiments repeat, or execute, the baseline experiment and each attack 120 times to compute the identification rate and the re-identification rates for the set of 120 source speakers. Execution $i$ of each attack with $i = 1, 2, \ldots , 120$ takes the de-identified utterance of $i \in S$ as input and Execution $i$ of the baseline experiment takes the corresponding original utterance as input. All other inputs (see also Section III) remain constant for all executions. In addition, as indicated in Section IV-A3, two variants of Attack 3 are executed: one using non-parallel (n-p) data and one using parallel (p) data for training the voice conversion function in the attack.

The corresponding identification and re-identification rates are shown in Fig. 1 and listed in Table I. In addition, the figure and the table show average rates for $N = 15, 30, 60$. These rates represent an average of the rates that would be computed if $N$ source speakers would have been selected at random from the set of 120 speakers while keeping all other settings the same as for $N = 120$. For each value of $N \in \{15, 30, 60\}$, 20 sets of $N$ speakers have been selected at random to compute the average rates shown in Fig. 1 and Table I, using the same sets of speakers for the baseline experiment and the three attacks. The figure and the table also show the expected re-identification rates of the attack based on random guessing for the different population sizes. Recall that this attack outputs one speaker from the source speaker set at random for each execution, and that its expected rate equals $1/N$. The results show that the identification rates of the baseline experiment are high for all population sizes. As indicated in Section III, Attack 1 is similar to the existing evaluation methods as used in [5], [9], [10]. In the experiments, Attack 2 significantly outperforms this known attack and the attack based on random guessing. Moreover, the rates of Attack 3 are higher than the rates of Attack 2; however, recall that Attack 3 uses a stronger attack model. Observe that the difference between the rates that are associated with parallel and non-parallel training data in Attack 3 is relatively large. Furthermore, note that the rates that are associated with parallel training data are close to the rates of the baseline experiment.

![Identification/re-identification rates](image)

**Fig. 1: Identification/re-identification rates**

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of source speakers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline experiment</td>
<td>0.99 0.98 0.96 0.96</td>
</tr>
<tr>
<td>Attack 3 (parallel)</td>
<td>0.92 0.92 0.88 0.86</td>
</tr>
<tr>
<td>Attack 3 (non-parallel)</td>
<td>0.46 0.40 0.31 0.26</td>
</tr>
<tr>
<td>Attack 2</td>
<td>0.40 0.31 0.24 0.19</td>
</tr>
<tr>
<td>Attack 1</td>
<td>0.24 0.15 0.12 0.04</td>
</tr>
<tr>
<td>Random guessing</td>
<td>0.07 0.03 0.02 0.01</td>
</tr>
</tbody>
</table>

**TABLE I: Identification/re-identification rates**
The setting for Attack 1 in [9] assumes that the target speaker is included in the source speaker set. This can cause the target speaker to be identified for a large number of re-identification attempts and consequently, a low re-identification rate. This can make the de-identification scheme look more effective than it is, since the attacker could first identify the target speaker in the source speaker set and then exclude this speaker from the set before performing Attack 1. In addition, after identifying the target speaker in the set, the attacker could use the utterance of the target speaker to mount Attack 3. For these reasons, the target speaker is not included in the source speaker set in the experiments in this paper. The rates of Attack 1 that are reported in other papers where the target speaker is excluded from the set, are similar to the rates of Attack 1 listed in Table I; however, it is generally difficult to directly compare results reported in different papers. Reasons for this are that different papers use different settings (such as different datasets and/or different amounts of training data) and that the use of different settings will typically impact re-identification rates.

V. CONCLUSION

Section III presented two new generic voice conversion attacks on speaker de-identification schemes. The first new attack assumes the same attack model as existing attacks and uses voice conversion to transform the de-identified utterance into utterances of all possible source speakers. The idea behind the attack is that a first source speaker - target speaker - second source speaker converted utterance is most similar to a second source speaker utterance if the first and the second source speaker are the same. The second new attack assumes a stronger attack model in which an utterance of the target speaker is known. The attack exploits this additional information to perform a de-identification process that is similar to the one performed by the designer, de-identifying utterances for all possible source speakers. The intuition behind this attack is that, among these utterances that are de-identified by the attacker, the one that is associated with the original source speaker will be most similar to the utterance that was de-identified by the designer.

The new attacks have been implemented using a GMM-based voice conversion method and a GMM-based identification scheme. Experiments were conducted to evaluate the security of GMM-based de-identification schemes and to compare the effectiveness of the new attacks with the effectiveness of existing attacks. The experimental results in Section IV show that the first new attack achieves re-identification rates that are significantly higher than the rates of existing attacks. In addition, the experiments show that the second new attack achieves even higher re-identification rates, exploiting the availability of an utterance of the target speaker. It was shown that these rates can be close to the identification rates of the baseline experiment if the attacker has access to parallel data for training the voice conversion function. For example, this will be the case if the designer uses a publicly available text-to-speech voice as the target speaker and if the attacker recognizes this voice.

A possible direction for future work is to investigate the effect of using different voice conversion methods for de-identification and the attacks, and/or the effect of using a different speaker identification scheme in the attacks. Additionally, one could investigate the effect of using different values of their parameters.

REFERENCES