Preventing Replay Attacks on Speaker Verification Systems

Conference Paper · September 2011

DOI: 10.1109/CCST.2011.6095943

2 authors:

Jesús Villalba
Johns Hopkins University
45 PUBLICATIONS **240** CITATIONS
SEE PROFILE

Eduardo Lleida
University of Zaragoza
211 PUBLICATIONS **1,249** CITATIONS
SEE PROFILE

Preventing Replay Attacks on Speaker Verification Systems

Jesús Villalba, Eduardo Lleida
Communications Technology Group (GTC),
Aragon Institute for Engineering Research (I3A),
University of Zaragoza, Spain
Email: {villalba,lleida}@unizar.es

***Abstract*—In this paper, we describe a system for detecting spoofing attacks on speaker verification systems. We understand as spoofing the fact of impersonating a legitimate user. We focus on detecting two types of low technology spoofs. On the one side, we try to expose if the test segment is a far-field microphone recording of the victim that has been replayed on a telephone handset using a loudspeaker. On the other side, we want to determine if the recording has been created by cutting and pasting short recordings to forge the sentence requested by a text dependent system. This kind of attacks is of critical importance for security applications like access to bank accounts. To detect the first type of spoof we extract several acoustic features from the speech signal. Spoofs and non-spoof segments are classified using a support vector machine (SVM). The cut and paste is detected comparing the pitch and MFCC contours of the enrollment and test segments using dynamic time warping (DTW). We performed experiments using two databases created for this purpose. They include signals from land line and GSM telephone channels of 20 different speakers. We present results of the performance separately for each spoofing detection system and the fusion of both. We have achieved error rates under 10% for all the conditions evaluated. We show the degradation on the speaker verification performance in the presence of this kind of attack and how to use the spoofing detection to mitigate that degradation.**

***Index Terms*—speaker verification, spoofing, forgery, replay attack, far-field, cut and paste.**

I. Introduction

Speaker recognition is the ability of recognizing people from their voices. Speaker recognition can refer to two fundamental tasks: verification and identification. Speaker verification is the task of determining whether a person is who he or she claims to be. In this case, possible imposters are not known to the system so it is an open-set task. Speaker identification is the task of determining who is talking among a known group of speakers. If we assume that the speaker must belong to that group of speakers it is called a closet-set identification. On the contrary, if the system must be able to say whether the speaker is none of the known speakers, the task includes identification and verification together and it is referred as open-set identification. We are interested in the application of speaker verification for *identity authentication* and *access control*. Speaker recognition can be used for controlling access to physical facilities [1], computer networks, web services or telephone resetting of passwords [2]. It can be especially important to increase safety of telephone banking transactions, electronic banking and e-commerce that have experienced an important growth in the recent years.

Current state of the art speaker verification systems (SV) have achieved great performance due, mainly, to the appearance of the GMM-UBM [3] and Joint Factor Analysis (JFA) [4] approaches. However, this performance is usually measured in conditions where impostors do not make any effort to disguise their voices to make them similar to any true target speaker and where a true target speaker does not try to modify his voice to hide his identity. That is what happens in NIST evaluations [5].

In this paper, we dealt with a type of attack known as spoofing. Spoofing is the fact of impersonating another person using different techniques like voice transformation or playing of a recording of the victim. There are multiple techniques for voice disguise. In [6] authors did a study of voice disguise methods and classified them into electronic transformation or conversion, imitation, and mechanical and prosodic alteration. In [7] an impostor voice is transformed into the target speaker voice using a voice encoder and decoder. More recently, in [8] an HMM based speech synthesizer with models adapted from the target speaker is used to deceive a SV system. In this work, we focus on detecting two kinds of spoofs: cut and paste and replay attack. These low technology spoofs are the most easily available to any impostor without speech processing knowledge.

The cut and paste spoofing attack can be applied to text dependent speaker recognition systems. In this kind of system, in the enrollment phase, the speaker is asked to utter a set of sentences several times (three utterances of two sentences in our tests). In the test phase, the speaker is asked to utter one of the sentences used for enrollment. The spoofing process consists of manufacturing the test utterance cutting and pasting fragments of speech (words, syllables) recorded previously from the speaker. We can decompose the spoofing detection task in two subtasks:

- Sentence selection: determining which one of the training sentences have been uttered in the test sentence.
- Cut and paste detection: detecting whether the test utterance has been made by concatenation of words. In this process we compare the test utterance with the training utterances of the selected sentence.

The far-field recording and replay attack can be applied to text dependent and independent speaker recognition systems. The

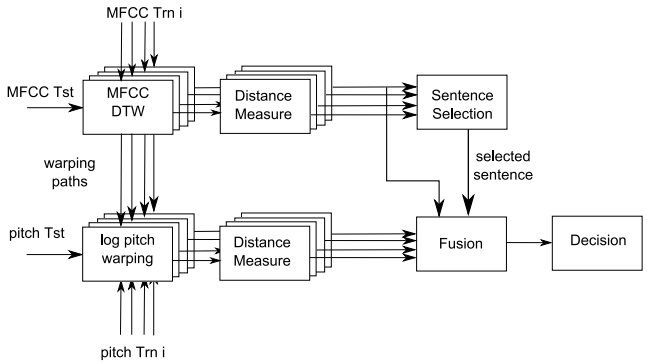

Fig. 1. Matching algorithm.

utterance used in the test is recorded by a far-field microphone and/or replayed on the telephone handset using a loudspeaker. These types of attacks can be used alone or jointly, which will make the detection task easier.

This paper is organized as follows. Section 2 and 3 explain the cut and paste and replay attack detection systems respectively. Section 4 describes the experiments and results. Finally, in section 5 we show some conclusions.

II. Cut and paste detection system

A. Features

The algorithm is based on DTW alignment of feature contours between training and test segments [9]. As features, we used:

- Logpitch: the fundamental frequency of speech is estimated with a pitch extractor based on the RAPT [10] implementation. That includes pitch tracking by dynamic programming.
- MFCC: 12 Mel Filtered Cepstral Coefficients (MFCC C1-C12) [11] are extracted. Mean and variance normalization of the MFCC (CMVN) is used to mitigate the channel mismatch effects on the results. Voice activity detection based on [12] is used to prune leading and trailing silence segments. We do not remove other silence segments.

We assume that these contours should be very different between a legitimate sentence and another one made of several recordings taken in different situations.

B. Matching algorithm

The template matching algorithm for detecting cut and paste using pitch and MFCC contours is summarized in Algorithm 1. Figure 1 shows a block diagram of our system.

In the next sections we explain more in detail each of the steps.

1) MFCC distance measure: The MFCC of the test segment is aligned with each one of the training sentences of the claiman speaker by DTW. Then, we calculate the Mahalanobis distance:

$$d_{MFCC}(r_i,t)^2 = \frac{1}{T}\sum_{k=1}^{T}\sum_{j=1}^{D}\frac{(rw_{ij}(k)-tw_j(k))^2}{\sigma_{w_j}^2} \quad (1)$$

where rw_i is the i^{th} warped reference signal, tw is the warped test signal, j is MFCC component index and k is the temporal index. We use the Mahalanobis distance to account for the different dynamic ranges of each of the MFCC components.

Algorithm 1 Cut and paste detection algorithm

Given a test sentence t and set of train sentences r_i.
for each pair of sentences (r_i, t) **do**
Perform join DTW alignment of the 12 MFCC of both sentences.
Use optimal warping path for warping the MFCC and log pitch contours.
Calculate Mahalanobis distance between the warped MFCC of both sentences:

- Use warped MFCC of the reference signal r_i for estimating the variances needed for calculating the Mahalanobis distance.
- Normalize de distance by the number of samples.

Calculate the Mahalanobis distance between the warped log pitch of both sentences:

- Calculate the distance using only the points where pitch have been detected on both signals.
- Detect possible halving and doubling pitch errors (pitch in one signal almost double than in the other).
- Use warped log pitch of the reference signal r_i for estimating the variances needed for calculating the Mahalanobis distance.
- Normalize de distance by the number of samples with pitch in both signals.

end for
Use the minimum MFCC distance to decide which of the allowed sentences has been uttered.
Fuse MFCC and log pitch distances by a weighted sum for the (r_i, t) pairs whose r_i matches the selected sentence.
Select the minimum of the fused distances as score.
Threshold score for decision.

2) Pitch Distance Measure: First, We warp the log pitch contour of the reference and test signals using the MFCC warping path. We found that missing pitch segments, in either the reference or the test signal, can produce big distances between pitch contours for non-spoof signals. As a result of this, the false acceptance rate (false spoof) could increase in an inevitable manner. The same could occur if there is halving or doubling pitch errors in any of the signals. If we try to align both pitch contours by DTW pitch errors could lead to a bad estimation of the correct warping path. This is the reason why we use the path estimated using the MFCC features to warp the pitch contour.

Even doing a good warping of the pitch contours, pitch errors degrade the performance of the system. To solve the missing pitch segments problem, we only use the frames that are voiced in the reference and test signals. Different sentences have very different number of voiced frames in common so normalizing the distance by the number of voiced frames is important.

For detecting halving and doubling errors, for each frame, we check if a doubled or halved version of the test pitch is almost the same as the reference pitch. Then we substitute the pitch value for the nearest one to the reference pitch. In mathematical notation:

$$\begin{aligned} &\exists n \in \mathbb{N} \ni rw(k) - tw(k) - n\log(2) < \epsilon \\ &\Rightarrow tw(k) = tw(k) + n\log(2) \end{aligned} \quad (2)$$

where $rw(k)$ is the reference warped log pitch and $tw(k)$ is the

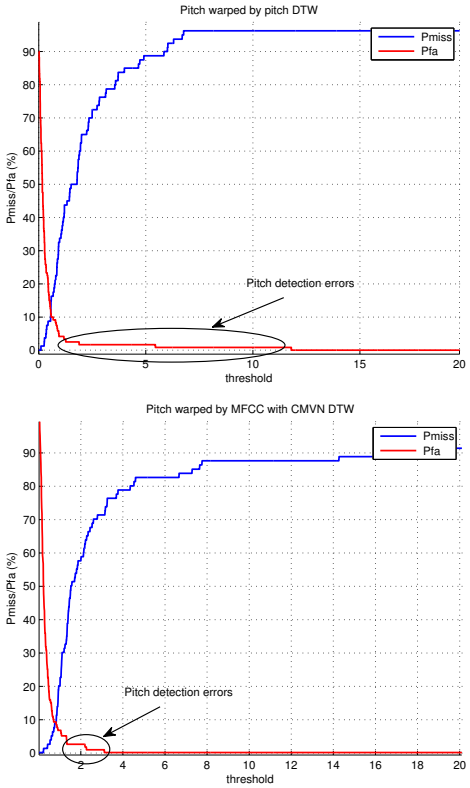

Fig. 2. Effect of pitch errors on error rates.

test warped log pitch

The effect of pitch detection errors can be seen clearly in Figure 2. The figure shows the curves of miss probability and false acceptance probability versus decision threshold for two systems based on pitch only. The left plot shows a system where the pitch has been warped using DTW between the train and test log-pitch contours. The right plot shows a system where the pitch has been warped using the path that we get doing DTW between the train and test MFCC contours. We can see that, for the left one, there is a probability of false acceptance that cannot be reduced but selecting a very high threshold. However, that would make it useless to detect any true spoofs. On the other hand, for the right one, the effect of pitch detection errors is not so harmful. That means that we can use our system in an operating point where we do not want to have false alarms and yet be able to detect a fair amount of spoofing attempts.

Once solved, as far as possible, the pitch detection errors we can proceed with the distance calculation:

$$d_{logpitch}(r_i,t)^2 = \frac{1}{|V|}\sum_{k\in V}\frac{(rw_i(k)-tw(k))^2}{\sigma_w^2} \quad (3)$$

$$V = \{k \ni rw_i(k) > 0 \wedge tw(k) > 0\} \quad (4)$$

where rw_i is the warped log pitch of the i^{th} reference signal, tw is the warped pitch of the test signal.

3) Sentence Selection: For determining which of the allowed sentences has been uttered by the user, we use the MFCC distance only. MFCCs are the typical features used in most speech recognition systems. According to our results, using MFCC is the most robust option to select the right sentence. We search for the training segment with minimum MFCC distance to the test segment. The sentence uttered in that segment is the selected sentence.

$$\hat{s}(t) = s\left(r_n\right) \ni n = \underset{i}{\operatorname{argmin}} d_{MFCC}(r_i,t) \quad (5)$$

where $s(f)$ is the sentence uttered in the segment f.

4) Fusion: We perform a weighted sum of MFCC and log pitch distances for the pairs (r_i,t) whose reference signal r_i has the selected sentence.

$$R = \{r_i \ni s(r_i) = \hat{s}\} \quad (6)$$

$$d(r_i,t) = w_1 d_{MFCC}(r_i,t) + w_2 d_{logpitch}(r_i,t) \ni r_i \in R\,. \quad (7)$$

The output score of the spoofing detection system is the minimum of these weighted distances:

$$d = \min_{r_i\in R} d(r_i,t)\,. \quad (8)$$

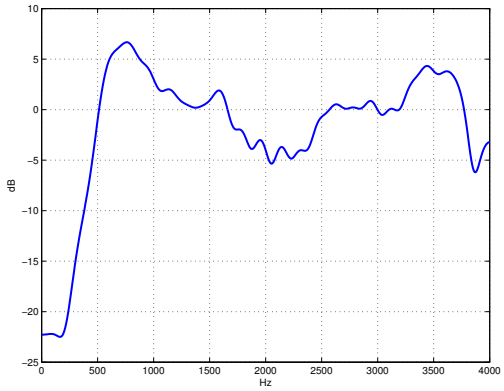

Fig. 3. Typical frequency response of smartphone loudspeaker.

III. Far field replay attack Detection system

A. Features

For each recording we extract a set of several features. These features were selected in order to be able to detect two types of manipulations on the speech signal:

- The signal have been acquired using a far-field microphone.
- The signal have been replayed using a loudspeaker.

Currently, speaker verification systems are mostly used on telephone applications. This means that the user is suppose to be near the telephone handset. If we can detect that the user was far from the handset during the recording we can consider it as an spoofing attempt. A far-field recording causes an increment of the noise and reverberation levels of the signal. This has as consequence a flattening of the spectrum and a reduction of the modulation indexes of the signal.

The simpliest way of injecting the spoofing recording into a phone-call is using a loudspeaker. Probably, the impostor would use a easily transportable device like a smart-phone, with a small loudspeaker. This kind of loudspeakers, usually, presents bad frequency responses in the low part of the spectrum. Figure 3 shows a typical frequency response of a smart-phone loudspeaker. We can see that the low frequencies, under 500 Hz, are strongly attenuated.

Following, we describe each of the features extracted.

Fig. 4. Modulation index calculation.

1) Spectral Ratio: The spectral ratio (SR) is the ratio between the signal energy from 0 to 2 kHz and from 2 kHz to 4 kHz. For a frame n, it is calculated as:

$$SR(n) = \sum_{f=0}^{NFFT/2-1} \log\left(|X(f,n)|\right) \cos\left(\frac{(2f+1)\pi}{NFFT}\right) \quad (9)$$

where $X(f,n)$ is the Fast Fourier Transform of the signal for the frame n. The average value of the spectral ratio for the speech segment is calculated using speech frames only. Using this ratio we can detect the flattening of the spectrum due to noise and reverberation.

2) Low Frequency Ratio: We call low frequency (LFR) ratio to the ratio between the signal energy from 100Hz to 300Hz and from 300Hz to 500Hz. For a frame n, it is calculated as:

$$LFR(n) = \sum_{f=100\text{Hz}}^{300\text{Hz}} \log\left(|X(f,n)|\right) - \sum_{f=300\text{Hz}}^{500\text{Hz}} \log\left(|X(f,n)|\right) \quad (10)$$

where $X(f,n)$ is the Fast Fourier Transform of the signal for the frame n. The average value of the low frequency ratio for the speech segment is calculated using speech frames only. This ratio is useful for detecting the effect of the loudspeaker on the low part of the spectrum of the replayed signal.

3) Modulation Index: The modulation index at time t is calculated as

$$Indx(t) = \frac{v_{max}(t) - v_{min}(t)}{v_{max}(t) + v_{min}(t)} \quad (11)$$

where $v(t)$ is the envelope of the signal and $v_{max}(t)$ and $v_{min}(t)$ are the local maximum and minimum of the envelope in a region close to the time t. The envelope is approximated by the absolute value of the signal $s(t)$ down sampled to 60 Hz. The average modulation index of the signal is calculated taken the frames whose index is above a threshold of 0.75. In Figure 4 we show a block diagram of the algorithm. The envelope of the far-field recording has higher local minimums due, mainly, to the additive noise. Therefore, it will have lower modulation indexes.

4) Sub-band Modulation Index: If the noise affects only to a small frequency band it may not have a noticeable effect on the previous modulation index. Thus, we calculate the modulation index on several sub-bands to be able to detect far-field recordings with coloured noises. The modulation index of each sub-band is calculated filtering the signal with a band-pass filter in the desired band previous to calculating the modulation index. We use indexes in the bands: 1kHz-3kHz, 1kHz–2kHz, 2kHz–3kHz, 0.5kHz–1kHz, 1kHz–1.5kHz, 1.5kHz–2kHz, 2kHz–2.5kHz, 2.5kHz–3kHz, 3kHz–3.5kHz.

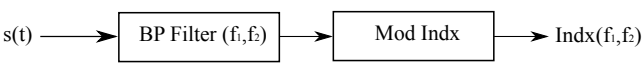

Fig. 5. Sub-band modulation index calculation.

B. Classification algorithm

Joining the features described in the previous section we get a feature vector for each recording:

$$x = (SR, LFR, Indx(0, 4kHz), \ldots, Indx(3kHz, 3.5kHz)) \ . \quad (12)$$

For each input vector x we apply the SVM classification function:

$$f(x) = \sum_i \alpha_i \mathrm{k}(x, x_i) + b \quad (13)$$

where k is the kernel function, and x_i, α_i and b are the support vectors, the support vector weights, and the bias parameter that are estimated in the SVM training process. The kernel that best suits our task is the Gaussian kernel.

$$\mathrm{k}(x_i, x_j) = \exp\left(\gamma \left\|x_i - x_j\right\|^2\right) \ . \quad (14)$$

We have used the LIBSVM toolkit [13]. For training the SVM parameters we have used data extracted from the training set of the SRE08 NIST database:

- Non spoofs: 1788 telephone signals of NIST SRE08 train set.
- Spoofs: synthetic spoofs made using interview signals from NIST SRE08 train set. We pass these signals through a loudspeaker and a telephone channel to simulate the conditions of a real spoof. We have used two different loudspeakers: a USB loudspeaker for a desktop computer and a mobile device loudspeaker; and two different telephone channels: analog and digital. In this way, we have 1475x4 spoof signals.

IV. Experiments

A. Database

There are not publicly available databases for this task so we have recorded our own one. The cut and paste database consists of two parts:

- P1: it has 20 speakers. It includes landline (T) signals for training, non spoof tests and spoofs tests; and GSM (G) for spoofs tests.
- P2: it has 10 speakers. It includes landline and GSM signals for all training and testing sets.

Each part has three sessions:

- Session 1: it is used for enrolling the speakers into the system. Each speaker has 3 utterances by channel type of 2 different sentences (F1,F2). Each sentence is about 2 seconds long.
- Session 2: it is used for testing non spoofing access trials and has 3 recordings by channel type of each of the F1 and F2 sentences.
- Session 3: it is made of different sentences and a long text that contain words from the sentences F1 and F2. They are recorded by a far-field microphone. The spoofing trials are created from the speech of this session. Several speech segments are extracted and used to build 6 recordings reproducing the sentences F1 and F2. After that, the signals are played on a telephone handset and transmitted through a landline or GSM channel. In this manner, these utterances include cut and paste and replay attack processing.

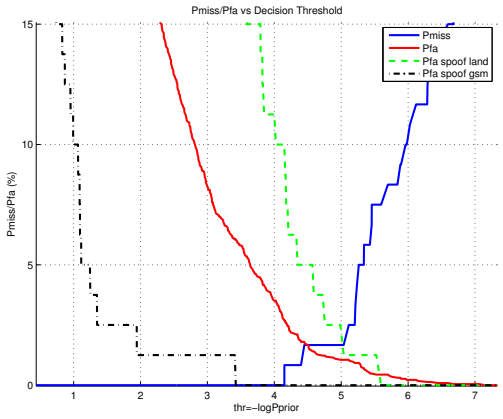

Fig. 6. Pmiss/Pfa vs decision threshold of database P1.

B. Speaker verification system

We used an SV system based on JFA [4] to measure the performance degradation. Feature vectors of 20 MFCCs (C0-C19) plus first and second derivatives are extracted. After frame selection, features are short time Gaussianized as in [14]. A gender independent Universal Background Model (UBM) of 2048 Gaussians is trained by EM iterations. Then 300 eigenvoices v and 100 eigenchannels u are trained by EM ML+MD iterations. Speakers are enrolled using MAP estimates of their speaker factors (y,z) so the speaker means super vector is given by $M_s = m_{UBM} + vy + dz$. Trial scoring is performed using a first order Taylor approximation of the LLR between the target and the UBM models like in [15]. Scores are ZT Normalized and calibrated to log-likelihood ratios by linear logistic regression using the FoCal package [16] and the SRE08 trial lists. We have used telephone data from SRE04, SRE05 and SRE06 for UBM and JFA training, and score normalization.

C. Speaker verification performance degradation

We did separate experiments using P1 and P2 datasets. For P1, we train speaker models using 6 landline utterances, and do 120 legitimate target trials, 2280 non spoof non target, 80 landline spoofs and 80 GSM spoofs. For P2, we train speaker models using 12 utterances (6 landline + 6 GSM), and do 120 legitimate target trials (60 landline + 60 GSM), 1080 non spoof non target (540 landline + 540 GSM) and 80 spoofs (40 landline + 40 GSM).

Using non spoof trials we have got and EER of 1.66% and EER of 5.74% for P1 and P2 respectively. In Figure 6 we show the miss and false acceptance probabilities against the decision threshold for P1 database. If we choose the EER threshold we have 5% of landline spoofs passing the speaker verification. None of the GSM spoofs would be accepted.

Figure 7 shows the score distributions for each of the databases. Table I shows the score degradation statistics due to the spoofing processing. The degradation is calculated by speaker and sentence type, that is, we calculate the difference between the average score of the clean sentences Fx of a given speaker and the average score of the spoofing sentences Fx of the same speaker. For P1, the spoofing scores are much lower than the true target scores but yet higher than the non target scores. For P2, the spoofing scores are lower than the non target scores. This means that the processing used for

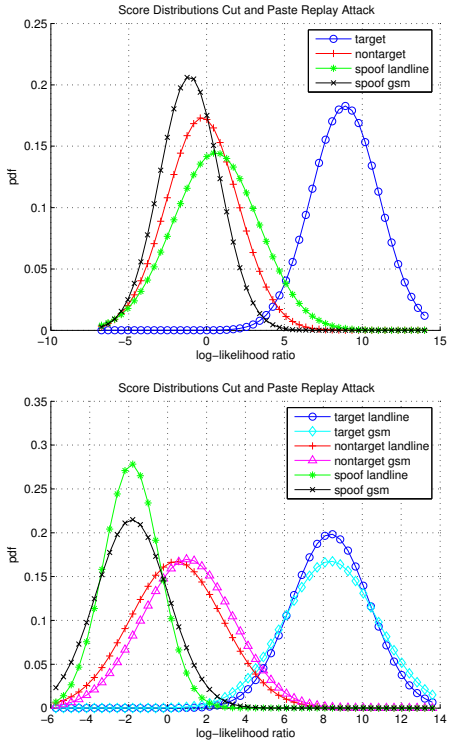

Fig. 7. Score distributions of databases P1 (top) and P2 (bottom).

creating the spoofs can modify the channel conditions in a way that makes the spoofing useless. We think that this is also affected by the length of the utterances. It is known that when the utterances are very short, Joint Factor Analysis cannot do proper channel compensation. If the channel component were well estimated the spoofing scores should be higher.

TABLE I
SCORE DEGRADATION DUE TO SPOOFING

			Mean	Std	Median	Max	Min
P1	T	Δscr	8.29	3.87	7.96	17.89	1.41
		Δscr/scr (%)	90.53	31.64	90.72	144.88	27.46
	G	Δscr	9.98	2.96	9.56	18.51	5.40
		Δscr/scr (%)	111.94	18.03	109.43	159.69	80.41
P2	T	Δscr	10.21	2.51	9.76	17.78	6.86
		Δscr/scr (%)	123.06	18.47	117.54	180.38	95.60
	G	Δscr	10.21	3.32	10.19	18.36	4.65
		Δscr/scr (%)	121.63	19.50	119.39	167.15	92.67

D. Cut and paste detection

We have done several experiments changing the type of telephone channels used in training, non spoof test and spoof test. In Table II we show EER for both databases for the different telephone channel conditions and features. The nomenclature used for defining each condition is: TrainChannel_NonSpoofTestChannel_SpoofTestChannel. Each channel can be: telephone landline (T), GSM (G), or a combination of both, as described in section IV-A. In this way, for example, condition T_T_TG means that we have used telephone landline as training and non spoofing data and compositions with replay attack transmitted by landline or GSM for spoofing data.

We see that MFCC produces better results than pitch. That is due to, on the one side, pitch detection errors, and on the

TABLE II
CUT AND PASTE DETECTION EER FOR MULTIPLE CHANNEL CONDITIONS.

EER(%)		MFCC12	Pitch	Pitch + MFCC	0.5 Pitch + 0.75 MFCC
P1	T_T_T	0.00	6.88	1.04	0.00
	T_T_G	0.00	3.12	0.00	0.00
	T_T_TG	0.00	6.46	0.73	0.00
P2	T_T_T	0.00	0.00	0.00	0.00
	T_G_G	0.00	0.00	0.00	0.00
	T_TG_TG	0.00	1.04	0.00	0.00
	G_T_T	0.00	2.92	0.00	0.00
	G_G_G	0.00	2.92	2.08	2.08
	G_TG_TG	0.00	3.96	1.04	1.04
	TG_T_T	0.00	2.92	0.00	0.00
	TG_G_G	0.00	0.83	0.00	0.00
	TG_TG_TG	0.00	2.50	0.00	0.00

TABLE III
SENTENCE SELECTION ERROR RATES FOR MULTIPLE CHANNEL CONDITIONS.

		$P(error)$	$P(error\|nonspoof)$	$P(error\|spoof)$
P1	T_T_T	13.00	0.00	32.50
	T_T_G	17.00	0.00	42.50
	T_T_TG	21.43	0.00	37.50
P2	T_T_T	17.00	0.00	42.50
	T_G_G	15.00	0.00	37.50
	T_TG_TG	16.00	0.00	40.00
	G_T_T	17.00	0.00	42.50
	G_G_G	10.00	0.00	25.00
	G_TG_TG	13.50	0.00	33.75
	TG_T_T	19.00	0.00	47.50
	TG_G_G	12.00	0.00	30.00
	TG_TG_TG	15.50	0.00	38.75

other side, that MFCC detects the channel mismatch due to the spoofing manufacturing process. We must remember that the original signals used to create the spoofs were first recorded by a far field microphone. Therefore, even for conditions like T_T_T there is some channel mismatch between spoofs and non spoofs. Despite that, we think that the most robust option is using a fusion of both features to be able to detect spoofs with small channel mismatch. This way we can cope with a wider range of situations that could happen in a real world application. Results are quite similar across conditions even when there is transmission channel mismatch between training and testing segments.

In Table III we show sentence selection error rates for each condition. The system is able to choose the correct sentence for non spoofing trials. This is important for not having detection of false spoofs. On the other side, the detection error rate is very high for the spoofing signals. This, far from being a problem, helps to increase the spoofing detection score.

E. Far field replay attack detection

In Table IV we show EER for both databases for the different channel combinations. The nomenclature used for defining each condition is: NonSpoofTestChannel_SpoofTestChannel. P1 database has higher error rates which could mean that they have been recorded in a way that produces less channel mismatch. That is also consistent with the speaker verification performance, the database with less channel mismatch has higher spoof acceptance. The type of telephone channel has little effect on the results. Figure 8 shows the spoofing detection DET curves.

TABLE IV
FAR-FIELD DETECTION EER

		EER(%)
P1	T_T	9.38
	T_G	2.71
	T_TG	5.62
P2	T_T	0.00
	G_G	1.67
	TG_TG	1.46

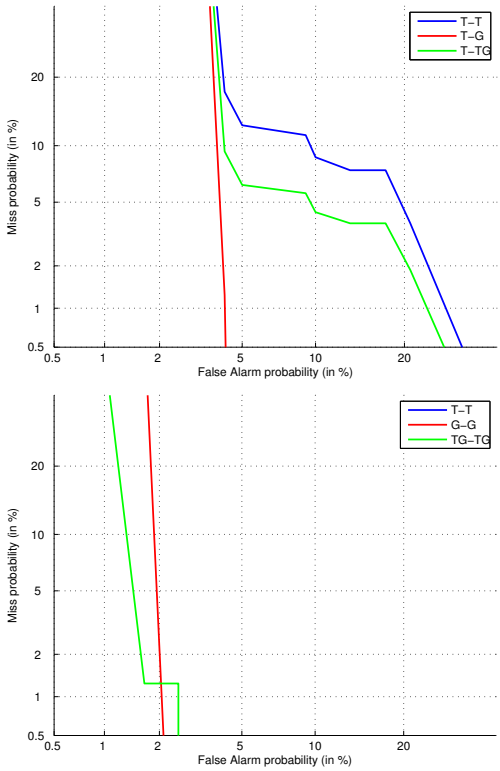

Fig. 8. DET far-field detection curves for P1 (top) and P2 (bottom).

F. Cut and paste and far field detection fusion

In Table V we show the results of the fusion of the cut and paste detector denoted as $0.5pitch + 0.75MFCC$ and the far field detector. For the fusion, we have used a weighted sum of both systems. Results are shown for each channel condition and for the pull of all conditions.

G. Fusion of Speaker Verification and Spoofing Detection

Finally we are going to fuse the spoofing detection and speaker verification systems. The fused system should keep similar performance to the original speaker verification system for legitimate trials but reduce the number of spoofing trials that deceive the system. We have done a hard fusion in which we reject the trials that are marked as spoof by the spoofing detection system; the rest of trials keep the score given by the speaker verification system. In order to not increase the number of misses of target trials, which would annoy the legitimate users of the system, we have selected a high decision threshold for the spoofing detection system.

We present results on the part P1 of the database because it has the higher spoofing acceptance rate. Figure 9 shows the

TABLE V
SPOOFING DETECTION EER FOR MULTIPLE CHANNEL CONDITIONS.

		EER(%)
P1	T_T_T	5.00
	T_T_G	0.62
	T_T_TG	3.75
	Pull	3.12
P2	T_T_T	0.00
	T_G_G	0.00
	T_TG_TG	0.00
	G_T_T	0.00
	G_G_G	0.00
	G_TG_TG	0.00
	TG_T_T	0.00
	TG_G_G	0.00
	TG_TG_TG	0.00
	Pull	0.00

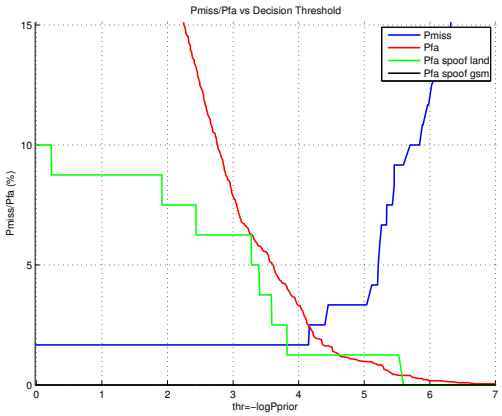

Fig. 9. Pmiss/Pfa vs. decision threshold for a speaker verification system with spoofing detection.

miss and false acceptance probabilities against the decision threshold for the fused system. If we again consider the EER operating point we can see that the number of accepted spoofs has decreased from 5% to 1.25% for landlines. Besides, all GSM spoofs are rejected no matter what SV decision threshold we choose. In exchange, we have to accept a minimum miss probability of 1.25% due to the false spoofs detected.

V. CONCLUSIONS

We have presented a spoofing detection system that combines replay attack and cut and paste detection. We have shown that by measuring distances between the pitch and MFCC contours of training and testing segments we can detect cut and paste attacks with low error rates. We have seen, too, that replay attacks change the spectrum and modulation indexes of the signal in a way that can be detected by a discriminative classifier. We have found that we can use synthetic spoofs to train the SVM model and yet, we can get good results on real spoofs. This method can significantly reduce the number of false acceptances when impostors try to deceive an SV system. This is especially important for persuading users and companies to accept using SV for security applications.

REFERENCES

[1] H. Gupta, V. Hautamaki, T. Kinnunen, and P. Franti, "Field Evaluation of Text-Dependent Speaker Recognition in an Access Control Application," in *Proceedings of the 10th International Conference Speech and Computer SPECOM2005*. Citeseer, 2005. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.144.4678\&rep=rep1\&type=pdf

[2] P. Roberts, "Visa Gets Behind Voice Recognition,," 2002. [Online]. Available: http://www.pcworld.com/article/106142/visa_gets_behind_voice_recognition.html

[3] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, Jan. 2000. [Online]. Available: http://dx.doi.org/10.1006/dspr.1999.0361

[4] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A Study of Interspeaker Variability in Speaker Verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 5, pp. 980–988, Jul. 2008. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4531370

[5] "NIST SRE10." [Online]. Available: http://www.itl.nist.gov/iad/mig/tests/sre/2010/NIST_SRE10_evalplan.r6.pdf

[6] P. Perrot, G. Aversano, and G. Chollet, "Voice disguise and automatic detection:review and perspectives," *Lecture Notes In Computer Science*, pp. 101–117, 2007. [Online]. Available: http://portal.acm.org/citation.cfm?id=1768226.1768233

[7] P. Perrot, G. Aversano, R. Blouet, M. Charbit, and G. Chollet, "Voice Forgery Using ALISP: Indexation in a Client Memory," in *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.* IEEE, 2005, pp. 17–20. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1415039

[8] P. L. De Leon, M. Pucher, and J. Yamagishi, "Evaluation of the vulnerability of speaker verification to synthetic speech," in *Proceedings of Odyssey 2010 - The Speaker and Language Recognition Workshop*, Brno, Czech Republic, 2010.

[9] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall PTR, May 2001. [Online]. Available: http://www.worldcat.org/isbn/0130226165

[10] D. Talkin, *A robust algorithm for pitch tracking (RAPT)*. Speech coding and synthesis, 1995, pp. 495–518.

[11] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1163420

[12] J. Ramirez, J. Segura, C. Benitez, A. de La Torre, and A. Rubio, "Voice activity detection with noise reduction and long-term spectral divergence estimation," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2004, pp. ii–1093–6. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1326452

[13] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001.

[14] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Oddyssey Speaker and*

Language Recognition Workshop, Crete, Greece, 2001. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.155.4456

[15] O. Glembek, L. Burget, N. Dehak, N. Brummer, and P. Kenny, "Comparison of scoring methods used in speaker recognition with Joint Factor Analysis," in *ICASSP '09: Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 4057–4060.

[16] N. Brummer, "Focal Bilinear." [Online]. Available: http://sites.google.com/site/nikobrummer/focalbilinear

Jesús Villalba received the degree in Telecommunication Engineering from University of Zaragoza (Spain) in 2004. Since he graduated he has been researching on speech and speaker recognition in the Group of Speech Technologies of the Aragon Institute for Engineering Research (I3A). He is doing his Ph.D. about robustness on speaker verification systems. He has leaded the I3A submissions to the NIST speaker recognition evaluations since 2006 achieving competitive results. He has done research internships in Brno University of Technology (BUT) and Agnitio Labs in South Africa. His current interests are speech quality measures, attacks to speaker verification systems and i-vectors based speaker verification.

Eduardo Lleida was born in Spain in 1961. He received the M.Sc. degree in telecommunication engineering and the Ph.D. degree in signal processing from the Universitat Politcnica de Catalunya (UPC), Barcelona, Spain, in 1985 and 1990, respectively. From 1986 to 1988, he was involved in his doctoral work at the Department of Signal Theory and Communications, UPC. From 1989 to 1990, he was an Assistant Professor and from 1991 to 1993, he was an Associate Professor in the Department of Signal Theory and Communications, UPC. From February 1995 to January 1996, he was a consultant in speech recognition with AT&T Bell Laboratories, Murray Hill, NJ. Currently, he is a Full Professor of signal theory and communications in the Department of Electronic Engineering and Communications, University of Zaragoza, Zaragoza, Spain, where he is heading a research team in speech recognition and signal processing. He is managing several speech-related project and he has coauthored more than 150 technical papers in the field of speech and speaker recognition, speech enhancement and recognition in adverse acoustic environments, acoustic modeling, confidence measures, and spoken dialogue systems.