SPEAKER VERIFICATION USING SECURE BINARY EMBEDDINGS

José Portêlo[12]*, Bhiksha Raj*[3]*, Petros Boufounos*[4]*, Isabel Trancoso*[12]*, Alberto Abad*[1]

[1] INESC-ID Lisboa, Portugal; [2] Instituto Superior Técnico, Lisboa, Portugal
[3] Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA
[4] Mitsubishi Electric Research Laboratories, Cambridge, MA, USA

ABSTRACT

This paper addresses privacy concerns in voice biometrics. Conventional remote speaker verification systems rely on the system to have access to the user's recordings, or features derived from them, and also a model of the user's voice. In the proposed approach, the system has access to none of them. The supervectors extracted from the user's recordings are transformed to bit strings in a way that allows the computation of approximate distances, instead of exact ones. The key to the transformation uses a hashing scheme known as Secure Binary Embeddings. An SVM classifier with a modified kernel operates on the hashes. This allows speaker verification to be performed without exposing speaker data. Experiments showed that the secure system yielded similar results as its non-private counterpart. The approach may be extended to other types of biometric authentication.

***Index Terms*—** Speaker verification, privacy, security

1. INTRODUCTION

Voice-based authentication systems, also often called speaker verification systems, have significant privacy concerns. Current systems require access to recordings of a user's voice. A malicious system, or a hacker who has compromised the system, could edit the recordings to impersonate the speaker. Even if the user only transmits features extracted from the voice, such that a recording cannot be synthesized from them, other risks remain. Information about the speaker's identity, gender, nationality, etc., could be deduced even from parameterized signals, and potentially abused. Also, there is scope for direct privacy violation. In order to authenticate a user, the system must retain a "model" for the user, which it can compare to incoming recordings. These models may now be used to uncover other recordings by the user, such as on services like YouTube, where the user may have assumed anonymity.

Our objective in this paper is to enable speaker verification in a manner that avoids risk to a user's privacy. To guarantee the privacy of the user, we require that the system should neither have access to the user's recordings, nor possess a model of the user's speech. These almost-paradoxical sounding requirements for a voice-based biometric system are not, in fact, unreasonable, and are assumed for other forms of secure biometrics as well [1]. The above requirements address our goals for the *system*; in addition we also desire that the system in turn must not be vulnerable to imposters who may gain access to a client device such as a smartphone that the user employs to connect to it. We will assume that all communication between the user and the system is over an appropriately secured channel, and that we need not to specifically consider either the man-in-the-middle or replay attacks.

Privacy concerns in voice biometric systems have been largely ignored until recently, and literature on the topic remains minimal. Pathak and Raj [2] treated the problem as one of secure function evaluation, employing homomorphic encryption methods to ensure that the system only sees encrypted data from the user, and only stores encrypted models that it cannot decrypt by itself, thereby satisfying privacy requirements. However, the computational overhead of repeated encryption and decryption makes this solution impractical. Moreover, the approach retains vulnerabilities to certain types of imposters. In [3] an alternate scheme based on Locality-Sensitive Hashing (LSH) was proposed, which converts voice recordings to password-like strings. Authentication is performed by matching the strings to stored templates. The method is fast and secure, satisfying both requirements mentioned above; however it compromises accuracy by requiring an *exact* match between hashes derived from the user's speech and those in the models stored by the system.

In this paper we propose a new privacy-preserving technique for speaker verification based on the recently proposed Secure Binary Embeddings (SBE) [4]. SBE is an LSH-like technique that converts vector data to bit strings, through a combination of random projections followed by banded quantization. This has specific properties that make it particularly useful in secure speaker verification. First, it has information theoretic guarantees of security, while having only the computational overhead of LSH. At the same time, it also permits us to compute Euclidean distances between vectors that are close enough by looking at the Hamming distance between the corresponding bit strings. This is an important characteristic of SBE hashes, as this way the perfect-match restriction of their LSH counterparts no longer applies. This means that classification tasks that rely on them are much less dependent

on the specific projections considered, therefore improving the overall performance.

The following section briefly presents the speaker verification algorithms we will secure. Subsequently, we describe secure binary embeddings and their guarantees in Section 3. Their application to privacy-preserving speaker verification is described in Section 4, together with some experiments demonstrating the efficacy of the proposed method. We follow it up with a discussion of actual usage scenarios and privacy issues in Section 5, and then present our conclusions.

2. SPEAKER VERIFICATION

In a conventional speaker verification system, the user provides the system with voice samples during an enrollment phase. The system employs these samples to build a "model" for the user. Later, incoming speech signals are compared to this model to verify the user.

State-of-art speaker verification techniques commonly employ a likelihood ratio test to perform verification. All speech signals are first parameterized into a sequence of feature vectors, typically mel-frequency cepstral coefficient (MFCC) vectors. As a first step, a large collection of recordings of non-target speakers is used to train a "Universal Background Model" (UBM). The UBM is a Gaussian mixture model (GMM) representing the distribution of speech from all potential imposters for the speaker. Subsequently, the UBM is adapted through *maximum a posteriori* adaptation[5] to the user's enrollment data to learn a GMM for the user. In addition to reducing enrollment data requirements, MAP adaptation also ensures a one-to-one correspondence between the Gaussians in the UBM and those in model for the speaker. Given a new recording purported to be from a target speaker, the log likelihood assigned to it by the GMM for the target speaker is compared to that obtained from the UBM to determine if the speaker must be accepted or not [5]. Many variations on this scheme have been proposed, primarily aimed at dealing with limited amounts of enrollment data, and mismatch between recording conditions in the enrollment and test data. These variants typically employ various flavors of factor analysis [6][7] to assign *a priori* probabilities to the parameters of the GMM for the speaker, and highly accurate authentication is reported under a variety of conditions.

An alternate equally-successful approach to likelihood ratio tests obtains a separate GMM for *each* of multiple enrollment recordings by the speaker, through MAP adaptation of the UBM to the recording. The parameters of the resulting GMM are concatenated into a "supervector" representing the recording. Supervectors are similarly obtained for recordings by putative imposters. A support vector machine (SVM) is then trained to distinguish between the two [8]. To verify that a given test recording was indeed spoken by the speaker, the supervector derived from the recording is classified by the SVM. Once again, *a priori* distributions may be assigned to the parameters through factor analysis – in this case the factor vectors may themselves be used to represent the recordings, and classification may be performed directly with them.

In this paper we employ the latter SVM-based approach to speaker verification. We use as a baseline classifier a conventional RBF kernel based SVM, where the kernel is given by $k(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \cdot d^2(\mathbf{x}_i, \mathbf{x}_j)}$. Here $d(\mathbf{x}_i, \mathbf{x}_j)$ refers to the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, and γ is a scaling factor. The RBF kernel has been observed to result in good classification accuracies for the speaker verification task [9]. Moreover, it is easily adapted to our privacy-preserving schemes.

3. SECURE BINARY EMBEDDINGS (SBE)

A *secure binary embedding* (SBE) is a scheme for converting real-valued vectors to bit sequences using band-quantized random projections. These bit sequences, which we will refer to as *hashes*, possess an interesting property: if the Euclidean distance between two vectors is lower than a threshold, then the Hamming distance between their hashes is proportional to the Euclidean distance between the vectors; if it is higher, then the hashes provide no information about the true distance between the two vectors. This scheme relies on the concept of Universal Quantization [10], which redefines scalar quantization by forcing the quantization function to have non-contiguous quantization regions.

Given an L-dimensional vector $\mathbf{x} \in \mathbb{R}^L$, the universal quantization process converts it to an M-bit binary sequence, where the m-th bit is given by

$$q_m(\mathbf{x}) = Q\left(\frac{\langle \mathbf{x}, \mathbf{a}_m \rangle + w_m}{\Delta}\right) \tag{1}$$

Here $\langle , \rangle$ represents a dot product. $\mathbf{a}_m \in \mathbb{R}^L$ is a projection vector comprising L i.i.d. samples drawn from $\mathcal{N}(\mu = 0, \sigma^2)$, Δ is a precision parameter, and w_m is a random dither drawn from a uniform distribution over $[0, \Delta]$. $Q(\cdot)$ is a quantization function given by $Q(x) = \lfloor x \bmod 2 \rfloor$. We can represent the complete quantization into M bits compactly in vector form:

$$\mathbf{q}(\mathbf{x}) = Q\left(\mathbf{\Delta}^{-1}(\mathbf{A}\mathbf{x} + \mathbf{w})\right) \tag{2}$$

where $\mathbf{q}(\mathbf{x})$ is an M-bit binary vector, which we will refer to as the *hash* of $\mathbf{x}$. $\mathbf{A} \in \mathbb{R}^{M \times L}$ is a matrix composed of the row vectors $\mathbf{a}_m$, $\mathbf{\Delta}$ is a diagonal matrix with entries Δ, and $\mathbf{w} \in \mathbb{R}^M$ is a vector composed from the dither values w_m.

The universal 1-bit quantizer of Equation 1 maps the real line onto $1/0$ in a banded manner, where each band is Δ_m wide. Figure 1 compares conventional scalar 1-bit quantization (left panel) with the equivalent universal 1-bit quantization (right panel).

The binary hash generated by the Universal Quantizer of Equation 2 has the following properties [4]: the probability that the i^{th} bits, $q_i(\mathbf{x})$ and $q_i(\mathbf{x}')$ respectively, of hashes of

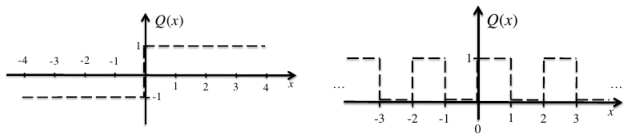

Fig. 1. 1-bit quantization functions.

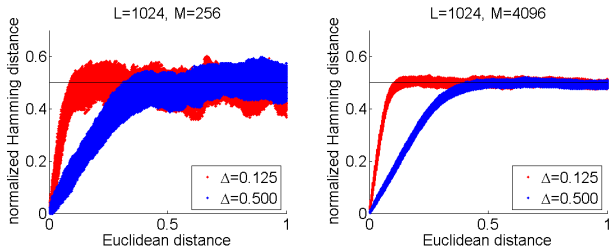

Fig. 2. SBE behavior as a function of Δ, for two values of M.

two vectors $\mathbf{x}$ and $\mathbf{x}'$ are identical depends only on the Euclidean distance $d = \|\mathbf{x} - \mathbf{x}'\|$ between the vectors and not on their actual values. As a consequence, the following relationship can be shown [4]: given any two vectors $\mathbf{x}$ and $\mathbf{x}'$ with a Euclidean distance d, with probability at most e^{-2t^2M} the normalized (per-bit) Hamming distance $d_H(\mathbf{q}(\mathbf{x}), \mathbf{q}(\mathbf{x}'))$ between the hashes of $\mathbf{x}$ and $\mathbf{x}'$ is bounded by:

$$\frac{1}{2} - \frac{1}{2}e^{-\left(\frac{\pi\sigma d}{\sqrt{2}\Delta}\right)^2} - t \leq d_H(\mathbf{q}(\mathbf{x}), \mathbf{q}(\mathbf{x}')) \leq \frac{1}{2} - \frac{4}{\pi^2}e^{-\left(\frac{\pi\sigma d}{\sqrt{2}\Delta}\right)^2} + t$$

where t is the control factor. The above bound means that the Hamming distance $d_H(\mathbf{q}(\mathbf{x}), \mathbf{q}(\mathbf{x}'))$ is correlated to the Euclidean distance d between the two vectors, if d is lower than a threshold (which depends on Δ). Specifically, for small d, $E[d_H(\mathbf{q}(\mathbf{x}), \mathbf{q}(\mathbf{x}'))]$, the expected Hamming distance, can be shown to be bounded from above by $\sqrt{2\pi^{-1}}\sigma\Delta^{-1}d$, which is linear in d. However, if the distance between $\mathbf{x}$ and $\mathbf{x}'$ is higher than this threshold, $d_H(\mathbf{q}(\mathbf{x}), \mathbf{q}(\mathbf{x}'))$ is bounded by $0.5 - 4\pi^{-2}exp\left(-0.5\pi^2\sigma^2\Delta^{-2}d^2\right)$, which rapidly converges to 0.5 and effectively gives us no information whatsoever about the true distance between $\mathbf{x}$ and $\mathbf{x}'$.

In order to illustrate how this scheme works, we randomly generated pairs of vectors in a high-dimensional space ($L = 1024$) and plotted the normalized Hamming distance between their hashes against the Euclidean distance between them (Figure 2). The number of bits in the hash is also shown in the figures. In all cases, once the normalized distance exceeds Δ, the Hamming distance between the hashes of two vectors ceases to provide any information about the true distance between the vectors. Changing the value of the precision parameter Δ allows us to adjust the distance threshold until which the Hamming distance is informative. Increasing the number of bits M leads to a reduction of the variance of the Hamming distance. A converse property of the embeddings is that for all $\mathbf{x}'$ except those that lie within a small radius of any $\mathbf{x}$, $d_H(\mathbf{q}(\mathbf{x}), \mathbf{q}(\mathbf{x}'))$ provides little information about how close $\mathbf{x}'$ is to $\mathbf{x}$. It can be shown that the embedding provides information theoretic security beyond this radius, if the embedding parameters $\mathbf{A}$ and $\mathbf{w}$ are unknown to the potential eavesdropper. Any algorithm attempting to recover a signal $\mathbf{x}$ from its embedding $\mathbf{q}(\mathbf{x})$ or to infer anything about the relationship between two signals sufficiently far apart using only their embeddings will fail to do so.

4. SPEAKER VERIFICATION WITH SBE

The application of the SBE to speaker verification systems is direct: if the classifier could be made to operate on SBE hashes of supervectors rather than on the supervectors themselves, speaker verification may be performed without exposing speaker data. The RBF kernel must be modified to work with Hamming distances between SBE hashes: $k(\mathbf{x}, \mathbf{x}') = e^{-\gamma \cdot d_H^2(\mathbf{q}(\mathbf{x}), \mathbf{q}(\mathbf{x}'))}$. Note that for a given $\mathbf{A}$ and $\mathbf{w}$, the modified kernel closely approximates the conventional RBF for small $d(\mathbf{x}, \mathbf{x}')$, but varies significantly from it at larger $d(\mathbf{x}, \mathbf{x}')$. While it does not satisfy Mercer's conditions and cannot be considered a true kernel, in practice it is effective as we shall see in the experiments below.

The implementation of a *privacy-preserving* speaker verification system is now as follows: the user communicates with the server through a smartphone or computation-capable device. In the enrollment phase, the supervectors for both the enrollment recordings and imposter recordings are computed by the user. Imposter recordings may be obtained from any public resource. The user computes SBE hashes from the supervectors and transmits them to the server. He retains the parameters $\mathbf{A}$ and $\mathbf{w}$ employed by the SBE as his private keys. The system trains an SVM with the obtained SBE hashes. During verification, the user computes the SBE hash for the supervector obtained from the test recording and transmits it to the system, which classifies it.

The system never sees the actual speech from the user. Its model for the user can only be used with the SBE hashes computed using the private hash key from that specific user, and is not usable without the user's participation. Moreover, an attacker wishing to pose as a specific user must not only manage to steal the embedding parameters from the user's client device but also gain access to voice recordings from the same user, as either of these alone are insufficient to gain unauthorized acces to the server. Therefore, our privacy requirements are satisfied.

4.1. Experiments using feature supervectors

As a proof of concept, we ran experiments on the YOHO Speaker Verification corpus [11], consisting of short utterances by 138 speakers. Each utterance contains a set of three two-digit numbers. The corpus is divided into two sets: enrollment and verification. The enrollment set (used for train-

#Gaussians	4	8	16	32	64	128
EER	3.55	1.52	0.60	0.25	0.22	0.21

Table 1. Speaker verification EER (%age), supervectors.

Δ	13.5	14.0	14.5	15.0	15.5
bpc=4	2.27	1.79	1.52	1.40	1.25
bpc=8	1.32	1.00	0.84	0.89	0.80
bpc=16	0.76	0.69	0.65	0.60	0.51

Table 2. Speaker verification EER (%age), SBE.

ing) contains 96 utterances from each speaker, totaling 14.54 hours of audio. The verification set (used for testing) contains 40 utterances from each speaker, totaling 6.24 hours of audio. We did not explicitly record imposters - instead for each of the 138 speakers in the corpus, the remaining 137 were used as imposters. The use of this corpus enables us to compare results with previous work on secure speaker verification [3]. The experiments used Gaussian mean supervectors based on MFCC features extracted in frames of 25ms, at the rate of 100 frames per second. For each frame we extracted 12 MFCC coefficients and the log-energy, augmenting them with the temporal differences and double-differences to result in a total of 39 features. A UBM was trained from the data for all the speakers. The UBM was adapted to each recording to obtain a single Gaussian supervector. The length of the supervectors depends on the number of Gaussians in the UBM: a UBM with N Gaussians results in supervectors with $L = 39N$ dimensions. In our baseline experiments, without SBE, we evaluated UBMs of different sizes, with the number of Gaussian components ranging from 4 to 128 Gaussians, to find the optimal settings. All experiments were performed using the LIBSVM toolkit [12]. Table 1 shows the results, averaging all the speakers, in terms of equal error rate (EER). The performance improves as the number of Gaussians increases. We do not present results with larger amounts (values up to 2048 Gaussians are common in the literature) because, for this particular corpus, they do not provide improvements. In fact, the results obtained with mixtures of 32 Gaussians are already very close to the ones obtained with 128. Hence, the experiments with SBE hashes will involve only 32 Gaussians.

4.2. Experiments using SBE hashes

The secure binary embeddings have two parameters that can be varied: the quantization step size Δ and the number of bits M. The value of M by itself is not a useful number, as different values of L (dimensionality of the supervector) require different values of M; hence we report our results as a function of *bits per coefficient* (*bpc*), computed as M/L. The *bpc* allows us to govern the variance of the universal quantizer. The results are presented in Table 2. As expected, both increasing Δ and *bpc* improves the classification performance. At *bpc*=16, the performance stabilizes rather quickly and thereafter is largely independent of Δ. Notice that we not only greatly improve on the 11.86% EER reported in [3], but also we produce an almost negligible increase in the classification error when compared with the non-secure version.

5. PRIVACY AND OTHER PRACTICAL ISSUES

Secure Binary Embeddings provide a basic but strong form of security: a vector $\mathbf{x}$ cannot be recovered, even in part, from its SBE $\mathbf{q}(\mathbf{x})$, if the projection matrix $\mathbf{A}$ and dither vector $\mathbf{w}$ are unknown. The primary benefit of using SBEs is that it now becomes possible for the system to perform classification using the SBEs $\mathbf{q}(\mathbf{x})$ without being able to recover the actual data $\mathbf{x}$ from it. Nevertheless, alternative factors that may provide information about the speaker must be considered. One of them is speaker *leakage*, which we define as the fraction of recordings from any speakers whose SBE hashes have a normalized Hamming distance below the threshold at which Hamming distance d_H is predictive of Euclidean distance d (which we empirically found to be 0.475 bits) with respect to any recording from another speaker. The left panel of Figure 3 shows how this varies with Δ. Not surprisingly, as Δ increases, this value increases; however at useful values of Δ this is very small. If the leaked vectors for any speaker show a bias towards specific other speakers, this would allow us to form speaker clusters. Ideally, the leaked vectors should be distributed uniformly across all other speakers, *i.e.*, the entropy of the distribution of the leaked vectors over imposters must be high. The right panel shows the normalized entropy of the distribution over imposters of the leaked vectors for each of the speakers in our test set, at a setting of Δ that results in 50% speaker leakage. The values obtained for the normalized entropy of each speaker are very high, with average values of 0.884, 0.862 and 0.851 for *bpc* values of 4, 8 and 16, respectively (1 represents completely random behavior; 0 indicates neighborhood to a single speaker). In other words, for the operational values of M and Δ, the identifiable bias of any speaker towards any other speaker is very low. Thus even if the system has registration data from a speaker, it must retrieve a very large number of putative recordings from a target speaker to make any inferences about other speakers in its database. However, this does not provide a strong guarantee of privacy against the motivated adversary.

6. CONCLUSIONS AND FUTURE WORK

The paper described a secure speaker verification approach which yields similar results to the non-secure counterpart and are a great improvement over previous results on the same task. The computational overhead is a very small price to pay for privacy. In order to extend the scheme to more sophisticated classifiers and to other forms of biometric authentication, several issues must be investigated. The nature of the

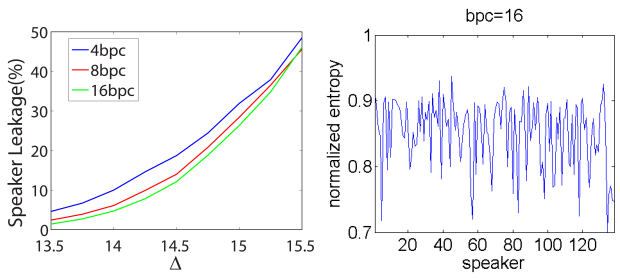

Fig. 3. Left: Speaker leakage as a function of Δ, for $bpc = 4$, 8 and 16. Right: Normalized entropy of the speaker leakage.

embedding restricts the form of classifiers that may be employed to those that utilize ℓ_2 distances. We are currently not only evaluating ways to extend the proposed work to use other forms of embeddings, but also analyzing mechanisms for more secure embeddings, as well as formally proving the non-invertibility of SBEs. Additionally, we also plan to further extend our work to use i-vectors [13] instead of supervectors, as well as considering other corpora [14] [15].

Acknowledgements

José Portêlo and Isabel Trancoso were supported by FCT grants SFRH/BD/71349/2010, PTDC/EIA-CCO/122542/2010 and PEst-OE/EEI/LA0021/2013. Bhiksha Raj was partially supported by NSF Grant 1017256. Petros Boufounos is fully supported by Mitsubishi Electric Research Laboratories.

7. REFERENCES

[1] A. Adler, "Biometric System Security", in *Handbook of Biometrics*, A.K Jain, P. Flynn and A. Ross Eds., Springer, 2007.

[2] M. Pathak and B. Raj, "Privacy Preserving Speaker Verification using adapted GMMs", in *Proc. Interspeech*, Florence, Italy, August 2011.

[3] M. Pathak and B. Raj, "Privacy-Preserving Speaker Verification as Password Matching", in *Proc. ICASSP*, Kyoto, Japan, March 2012.

[4] P. Boufounos and S. Rane, "Secure Binary Embeddings for Privacy Preserving Nearest Neighbors", in *Proc. Workshop on Information Forensics and Security (WIFS)*, Foz do Iguaçu, Brazil, December 2011.

[5] D. A. Reynolds, T. F. Quatieri and R. B. Dunn, "Speaker Verification using Adapted Gaussian Mixture Models", *Digital Signal Processing*, Volume 10, Issues 1-3, pp. 19–41, January 2000.

[6] P. Kenny, G. Boulianne, P. Ouellet and P. Dumouchel, "Joint Factor Analysis Versus Eigenchannels in Speaker Recognition", *IEEE Trans. Audio, Speech and Language Processing*, 15(4), pp. 1435–1447, 2007.

[7] M. Sennoussaoui, P. Kenny, N. Brummer, E. de Villiers and P. Dumouchel, "Mixture of PLDA Models in I-Vector Space for Gender-Independent Speaker Recognition", in *Proc. Interspeech*, Florence, Italy, August 2011.

[8] W. M. Campbell, J. R. Campbell, D. A. Reynolds, E. Singer and P. A. Torres-Carrasquillo, "Support Vector Machines for Speaker and Language Recognition", *Computer Speech and Language*, vol. 20, pp. 210–229, 2006.

[9] X. Anguera, "Minivectors: an Improved GMM-SVM Approach for Speaker Verification", in *Proc. Interspeech*, Brighton, United Kingdom, September 2009.

[10] P. Boufounos, "Universal Rate-Efficient Scalar Quantization", *IEEE Trans. on Information Theory*, 58(3): 1861–1872, 2012.

[11] J. P. Campbell, "Testing with the YOHO CD-ROM Voice Verification Corpus", in *Proc. ICASSP*, Detroit, Michigan, USA, May 1995.

[12] C.-C. Chang and C.-J. Lin, "LIBSVM : A Library for Support Vector Machines", *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[13] N. Dehak, R. Dehak, P. Kenny, N. Brümmer, P. Ouellet and P. Dumouchel, "Support Vector Machines versus Fast Scoring in the Low-Dimensional Total Variability Space for Speaker Verification", in *Proc. Interspeech*, Brighton, United Kingdom, September 2009.

[14] A. Martin and C. Greenberg, "The NIST 2010 Speaker Recognition Evaluation", in *Proc. Interspeech*, Makuhari, Japan, September 2010.

[15] R. Woo, A. Park and T. Hazen, "The MIT Mobile Device Speaker Verification Corpus: Data Collection and Preliminary Experiments", in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, San Juan, Puerto Rico, June 2006.