ROBUST SPEECH HASHING

by

Ekin Olcan Şahin

B.S., in Electrical and Electronics Engineering, Bilkent University, 2006

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Department of Electrical and Electronics Engineering

Boğaziçi University

2009

ACKNOWLEDGEMENTS

I would like to thank:

M. Kıvanç Mıhçak

Murat Saraçlar, S. Serdar Kozat, Bülent Sankur

Özgur Dalkılıç

Deniz Çelebi

Family

Erinç Dikici, Doğan Can, Ebru Arısoy

Techneon, BÜSİM

ABSTRACT

ROBUST SPEECH HASHING

In this thesis, a robust speech hashing algorithm is proposed and performance of this speech hashing algorithm is compared with several robust audio hashing algorithms. We use phone based frequency-time domain analysis for developing a fingerprint(hash value) for any speech data. Robust Speech Hashing can qualitatively be stated as a "dimensionality reduction" mechanism (which would be called the "robust speech hashing" function) via which the desired content of interest can be tracked and found reliably. Phonemes as speech characteristics and randomized frequency as hashing backbone is used so as to conclude on a secure speech tracking. The proposed algorithm is formed by 3 basic stages: Offline, Online and Comparison stages are applied in order. First, we extract most effective letter patterns in the cepstral domain. After transforming the speech signal into the spectral domain, the cepstrum coefficients are projected on the subspace spanned by the pattern that represents the letter (vowel) at hand. Moreover a pseudo-random linear transformation is applied in order to add a secure aspect. Lastly, the robust hash values of audio files are compared in the L_2 sense. The comparison takes place between between different audios as well as same but attacked ones. Several comparison tests are made for robust speech hash value based identification. ROC curves for different kind of attacks are investigated and we determined that, for speech signals, the proposed algorithm is superior to other considered robust audio hashing functions.

ÖZET

GÜRBÜZ SES KIYIM FONKSİYONU

Bu çalışmada, yeni bir gürbüz ses kıyım algoritması sunulmuş ve farklı gürbüz ses kıyım fonksiyonları ile performans karşılaştırması yapılmıştır. Konuşma verileri için parmakizi (kıyım değeri) ses bazlı frekans-zaman uzayı analizi yapılarak bulunmaktadır. "Gürbüz Konuşma Kıyım"ı içerik takibi yapmak ve bulmak için tasarlanmış bir boyut düşüren mekanizma olarak görülebilir. Güvenilir konuşma verisi takibi için, konuşma sesinin özelliği düşünülerek fonem (en küçük ses değeri) kullanılmış ve kıyım fonksiyonu omurgası olarak rasgele yoğunluk hesaplanmıştır. Sunulan algoritma 3 temel aşamadan meydana gelmektedir: sırasıyla çevrimdışı, çevrimiçi ve karşılaştırma aşamaları uygulanmaktadır. İlk olarak, en yoğun kullanılan sesli harfin zaman-frekans uzayında örüntüsü çıkarılmıştır. Daha sonra değerlendirilecek ses sinyali zaman-enerji ekseninden zaman-frekans eksenine çevrilmiş ve, "cepstrum" katsayılarının, sesli harfin örüntüsünün kapsadığı altuzaya izdüşümleri hesaplanmıştır. Bununla beraber, güvenlik konusu sözde-rasgele doğrusal dönüşüm uygulanarak halledilmiştir. Son konuşma verilerinden hesaplanmış gürbüz ses değerlerinin L_2 uzaklıkları karşılaştırılmış, karşılaştırma hem aynı hem farklı konuşma değerlerinin orijinal ve saldırılmış sürümleri arasında yapılmıştır. Kıyım değeri esas alınarak tanıma yapmak için bir çok test yapılmış alıcı işletim eğrileri incelenmiştir. Çıkan sonuçları değerlendirdiğimizde önerilen algoritmanın değerlendirilen diğer ses kıyım fonksiyonlarından -konuşma sesleri için- daha iyi olduğunu görürüz.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZET v
LIST OF FIGURES vii
LIST OF SYMBOLS/ABBREVIATIONS ix
1. INTRODUCTION 1
1.1. On the Need of Robust Speech Hashing 4
1.2. Background and Prior Art 6
2. PROPOSED SPEECH HASHING APPROACH 14
2.1. Notation 14
2.2. Proposed Speech Hashing Algorithm 14
2.2.1. Offline Pattern Extraction 16
2.2.2. Online Hash Extraction 20
2.2.3. Hash Based Perceptual Comparison 23
3. EXPERIMENTAL RESULTS AND DISCUSSION 24
3.1. Potential Developments 31
4. CONCLUSION 33
REFERENCES 35

LIST OF FIGURES

Figure 2.1. Proposed Speech Hashing Algorithm Overall Flow. 15

Figure 2.2. Offline Phase: Spectral Pattern Extraction 19

Figure 2.3. Online Phase: Hash value computation 22

Figure 3.1. ROC under Additive White Gaussian Noise. 26

Figure 3.2. ROC under Back Ground Noise. 26

Figure 3.3. ROC under 10dB boost attack 27

Figure 3.4. ROC under 10dB cut. 27

Figure 3.5. ROC under heavy echo attack. 27

Figure 3.6. ROC under non linear distortion attack. 28

Figure 3.7. ROC under heavy non linear distortion attack. 28

Figure 3.8. ROC under heavy notch filter attack. 28

Figure 3.9. ROC under light notch filter attack. 29

Figure 3.10. ROC under old time radio attack. 29

Figure 3.11. ROC under change in high energy values. 29

Figure 3.12. ROC under heavy pitch lower attack. 30

Figure 3.13. ROC under pitch raise. 30

Figure 3.14. ROC under change in pitch of 2 different wave. 30

LIST OF SYMBOLS/ABBREVIATIONS

a_{ij}	Description of a_{ij}
α	Description of α
DA	Description of abbreviation
L	Number of training signals
$\mathbf{s}_i$	the i-the training digital signal, $1 \leq i \leq L$
α	a specific chosen letter of interest that is used during the offline and online phases of the algorithm; α would be an element of the Turkish alphabet
M_i^α	number of places, where a chosen letter of α is observed within a training signal of $\mathbf{s}_i$, $1 \leq i \leq L$
$\tilde{\mathbf{p}}_{j,i}^\alpha$	a vector that represents the central part of the tri-phone representation of a chosen letter α, within the training signal $\mathbf{s}_i$ at the location j, where $1 \leq j \leq M_i^\alpha$, $1 \leq i \leq L$
N_{FFT}	the size of FFT that is being applied during the offline and online phases of the algorithm
$\tilde{\mathbf{q}}_{j,i}^\alpha$	the length-N_{FFT} magnitude-FFT of the vector $\tilde{\mathbf{p}}_{j,i}^\alpha$
$\hat{\mathbf{q}}^\alpha$	the empirical mean of $\left\{\tilde{\mathbf{q}}_{j,i}^\alpha\right\}_{i,j}$
$\mathbf{x}$	a digital input speech signal to the robust speech hashing algorithm
N	length of $\mathbf{x}$
$\mathbf{x}_i$	the i-th "time block" of the signal $\mathbf{x}$
N_{blocks}	number of time blocks within signal $\mathbf{x}$; i.e., $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_{blocks}}]$
$\mathbf{u}_i$	the length-N_{FFT} magnitude-FFT of the vector $\mathbf{x}_i$
K	the secret key, which is used as the seed of a secure PRNG that is utilized during the online phase of the hash algorithm
N'	the length of the output hash vector, which is also equal to the number of regions used during the online phase of the algorithm

$\mathcal{R}_j$	the j-th pseudo-randomly-generated "region" that is used during the online phase of the algorithm, where $\mathcal{R}_j = \{A_j, A_j + 1, \ldots, B_j - 1, B_j\} \subseteq \{1, 2, \ldots, N_{blocks}\}$. Here, for each j, A_j and B_j are appropriately pseudo-randomly generated parameters that uniquely define the region $\mathcal{R}_j$ and $1 \leq j \leq N'$
$\mathbf{w}_j$	the j-th Gaussian smoothly-varying pseudo-random weight vector that is defined to calculate the j-th component of the hash vector. Here, for each j, $\mathbf{w}_j \in \mathbb{R}^{L_j}$, where L_j represents the length of $\mathcal{R}_j$, i.e., $L_j = B_j - A_j + 1$
$\mathbf{h}_K^\alpha$	the output hash vector, of length-N', of the digital speech signal $\mathbf{x}$, which is computed using the secret key K and the input letter α

1. INTRODUCTION

It is an undeniable fact that in today's modern era, there is an increasing trend toward expansing of digital data. There is a particular emphasis on distribution and sharing of digital movies, images, audio clips and speech data. Such data are excessively being used for personal leisure as well as professional purposes, such as training and education. While a common data distribution mechanism remains to be P2P networks among end users, many corporations and institutions are also promoting and utilizing schemes for professional training and marketing [1], including concepts of "E-Learning" and "M-Learning" (the former refers to dissemination of professionally valuable multimedia data over TCP/IP, mainly via e-mail, whereas the latter refers to the delivery of valuable information over mobile devices such as PDA's, cell phones and laptops).

There are several issues that need to be taken into account while designing and developing such distribution mechanisms, ranging from automatic adjustment of quality of service to protection of data, thereby ensuring customer satisfaction and system security. In my thesis, I particularly plan to focus on the security aspects of digital multimedia data distribution, with an emphasis on speech. Conventional data protection mechanisms include usage of several cryptographic primitives, resulting in the framework of "Digital Rights Management" (DRM). Primitives such as stream ciphers, public key encryption and digital signatures are commonly used to monitor and control accessibility by various users and/or user groups (such as [2]). In addition to these techniques, recent multimedia protection mechanisms can potentially be extremely useful in maintaining data security in the presence of "analog hole" (i.e., when the multimedia data can no longer be traced using purely digital precautions, such as cryptography, but rather data need to be protected based upon decisions on its perceptual content).

As far as protection of multimedia data against piracy goes, there are two fundamental trends: robust signal hashing and robust signal watermarking (either for screening or traitor tracing purposes, where the latter is also known as "fingerprinting" in the multimedia security literature) [3].

If we consider watermarking as a security mechanism, signals are modified at hand in order to provide content protection. But you cannot modify the signals that are already put out to the market; that is the major problem in most practical watermarking applications. Therefore, In this thesis, I would like to focus particularly on robust speech hashing. The main idea is to extract robust (i.e., approximately invariant under malicious modifications) and secure (i.e., produced as a result of a secure pseudo-random algorithm where the key of the pseudo-random number generator is used as the secret) features, which in return are used to automatically detect any leaked data. This aspect will be discussed in more detail. For further information on robust signal hashing that can be applied to speech signals, we refer the interested reader to [4]. In addition, robust signal hashing can also be used to achieve fast data retrieval within a speech database.

A hash function is usually computationally efficient function that maps long inputs to shorter outputs. For an input sequence I of length N, the result, hash value is a binary sequence of length h ($\vec{h} = H_K(I)$); where N >>h. The uses of hash functions are many and indeed wide-ranging: content tracking, content authentication, anti-piracy search, compilers, checksums, different searching and sorting algorithms, cryptographic message authentication, one-way hash functions for digital signatures, time stamping, etc.[4]. Conventional Hash Functions use random keys in order to produce hash values satisfying randomness and pair wise independence properties. The produced hash values must be uniformly distributed and must be statistically independent of each other. They use some random seeds (keys) and seek these goals. More explicitly;

1. **Randomization** : For any given input I, its hash value should be approximately uniformly distributed among all possible 2^h outputs (hash values are h bits long).

$$\forall h \in \{0,1\}^h, \quad Pr\{H_K(I) = \vec{h}\} \approx 2^h. \tag{0.1}$$

2. **Pairwise Independence** : The hash outputs for two perceptually different in-

puts (say I_1 and I_2) should be approximately independent:

$$\forall h_1, h_2 \in \{0,1\}^h, Pr\{H_K(I_1) = \vec{h_1} | H_K(I_2) = \vec{h_2}\} \approx Pr\{H_K(I_1) = \vec{h_1}\}. \quad (0.2)$$

It is obvious that conventional hash functions are one-to-one, which is not desired in multimedia applications. Multimedia data can be found in many different formats and multimedia content may undergo several changes (like AD/DA conversion, reduction of quality, compression, additional noise, signal processing algorithms). Therefore, same hash values should be computed for two originally same contents, which are found in two different formats or which may be modified within acceptable range modifications. This idea leads to the additional constraint, which is perceptual similarity.

- **Perceptual similarity** For all possible acceptable disturbances, the output of the hash function should remain approximately invariant. Two audio clips are defined as perceptually similar if they sound the same, which should give the same hash values or the hash values should be close in some distance sense. Let I and $\hat{I}$ be perceptually similar signals.

$$\forall h \in \{0,1\}^h, \quad Pr\{H_K(I) = \vec{h}\} \approx Pr\{H_K(\hat{I}) = \vec{h}\}. \quad (0.3)$$

We propose an algorithm to achieve these goals.

For instance, as a real-time application, we may consider a deal with an artist and Microsoft. The artist is willing to sell his/her content via Internet, while being protected against piracy. The deal is as follows: Microsoft will get a commission from each piece that is sold, while tracking the leaked data. The leaked data may be found in several different formats, reduced in size, may undergo in several signal processing algorithms, therefore there should be a perceptual search algorithm. As already mentioned, perceptual search algorithms are important applications of robust signal hashing. The tracking procedure will be as follows: - preprocess: Hash values will be extracted from each artist's piece - preprocess: Hash values will be extracted from each piece that is found over internet or some databases - search: The hash values

will be compared (in the sense of Euclidean norm)

So far, most of the effort on the design of practical robust signal hashing systems has been focused on robust image hashing, robust audio hashing and robust signal hashing for biometrics. [4],[5],[6]. In this thesis, we focus on robust signal hashing, within a fundamentally different context, that is within the context of speech data.

1.1. On the Need of Robust Speech Hashing

Robust Multimedia Hashing is binary representatives for multimedia input sources such that hash values satisfy two requirements: they remain invariant under a constrained set of attacks, and they are different for perceptually different inputs with high probability [4].Clearly, multimedia hash function is many-to-one mapping.

Multimedia consists audio and visual data. Several approaches have been considered for both image hashing[7],[5],[8] and audio hashing[9],[10],[11]. Audio hashing techniques are usually studied with spectral characteristics. After transforming into a time-frequency domain audio is analyzed as an image. None has ever consider the signal-based characteristics for audio data. Especially there is no explicit study for speech signal.

Valuable speech data present in digital environment that's why speech hashing is necessary. We may provide a couple of examples:

- Example 1: There are language courses which are available on market. While commuting to work, running errands, taking a trip, driving or even sleeping such digital courses may offer a great improvement and ease for both basic and advanced learners. These speech signals that are useful to learn languages are obviously valuable data. Until now, valuable speech data are treated as an audio signal for anti-piracy protection. Language courses' files have high signal-to-noise ratio(SNR), so it will not be difficult to output a clear and definite hash value. But the leaked versions have low SNR, where conventional audio hash algorithms

may be improved.

- Example 2: An online lecture is an educational lecture prepared to be available online. Lectures are recorded to video, audio or both, and then uploaded on a web site(our concern is audio). Lectures are accessible by students or authorized persons. Although the setup seems to be well protected, there may be leakage at any time, because of the database software or the users. Lectures do not have many different speakers, do not have a colored, informative spectral characteristics. Therefore hash algorithms may be improved by adding some other stuff.

In such cases, sufficiently long speech signals are of significant value and thus form economically valuable commodities. This is precisely why it is of interest to develop and research content protection methods and approaches geared toward such speech signals.

Attackers may somehow get hold of the speech signal of value and try to sell it in the black market or perhaps freely distribute it in P2P networks. Note that in such anti-piracy applications pirates potentially modify the speech signal of interest while still preserving the perceptual quality, which, in turn, implies that the desired content protection mechanisms should be designed such that they are robust to such modifications. Modifications may be hand-made or made automatically but the content would not be changed radically so as to keep the value or the content quality. Type-change, reduction in dimensionality, noise, packet loss, some signal processing algorithms and DA/AD converters may be counted as these kind of attacks. It is obvious that the content is still same, but there is a change in digital details.

All in all, the final goal, can qualitatively be stated as to come up with a "dimensionality reduction" mechanism (which would be called the "robust speech hashing" function) via which the desired content of interest can be tracked and found reliably.

1.2. Background and Prior Art

We categorized Robust signal Hashing into three factions for simplicity. First one is the robust image and video hashing. Second one is the robust audio hashing and lastly other than anti-piracy robust signal hashing. Here we provide an overview of the leading existing robust signal hashing approaches.

Category 1:Robust image and video hashing prior study

- Venkatesan - Koon - Jakubowski - Moulin - ICIP 2000 - Robust Image Hashing paper This approach has built for managing large image databases. It is for sure that rapid growth of digital images creates problems for managing, indexing and protection. This paper proposed that image hash functions may be used as image indexing technique in such cases. The algorithm uses randomized signal processing strategies for a non-reversible compression of images into random binary strings, and is shown to be robust against image changes due to compression, geometric distortions, and other attacks. This algorithm brings to images a direct analog of Message Authentication Codes (MACs) from cryptography, in which a main goal is to make hash values on a set of distinct inputs pairwise independent. This minimizes the probability that two hash values collide, even when inputs are generated by an adversary.
- Mihcak - Venkatesan - SPDRM - Iterative Geometric Hashing Here, proposed algorithm put forward a novel and robust hashing paradigm that uses iterative geometric techniques and relies on observations that main geometric features within an image would approximately stay invariant under small perturbations. This algorithm is to produce sufficiently randomized outputs which are unpredictable, thereby yielding properties akin to cryptographic MACs. This is a key component for robust multimedia identification and watermarking (for synchronization as well as content dependent key generation). The hashing technique is relatively media independent and works for audio as well.
- Kozat Venkatesan Mihcak SVD Based Robust Image Hashing.
  Images (as well as attacks on them) may be viewed as a sequence of linear oper-

ators and hashing algorithm that is proposed by Kozat,Mihcak,Venkatesen [12] employs transforms that are based on matrix invariants. To derive the linear operator sequence, a two dimensional representation of an image by a sequence of (possibly overlapping) rectangles R_i whose sizes and locations are chosen randomly [13] from a suitable distribution. The restriction of the image (representation) to each R_i gives rise to a matrix A_i.

The algorithm first constructs a secondary image, derived from input image by pseudo-randomly extracting features that approximately capture semi-global geometric characteristics. From the secondary image (which does not perceptually resemble the input), the final features are extracted which can be used as a hash value (and can be further suitably quantized). In this paper, spectral matrix invariants as embodied by singular value decomposition are used. Formation of the secondary image turns out be quite important since it not only introduces further robustness (i.e., resistance against standard signal processing transformations), but also enhances the security properties (i.e. resistance against intentional attacks).

- Monga - Mihcak - NMF Based Robust Image Hashing - IEEE TIFS 2008 Journal paper In this paper, the use of non-negative matrix factorization (NMF) for robust image hashing is studied. In particular, images are treated as matrices and the goal of hashing is a randomized dimensionality reduction that retains the essence of the original image matrix while preventing against intentional attacks of guessing and forgery. Standard-rank reduction techniques such as the QR, and singular value decomposition (SVD), produce low rank bases which may omit the structure (i.e. non-negativity for images) of the original data. Therefore, it is observed that NMFs have two very desirable properties for secure image hashing applications:
  - The additivity property resulting from the non-negativity constraints results in bases that capture local characteristics of the image, thereby significantly reducing misclassification.
  - The effect of geometric attacks on images in the spatial domain reveals approximately as independent identically distributed noise on NMF vectors, allowing design of detectors that are both computationally simple and at

the same time optimal in the sense of minimizing error probabilities.

- Oostveen - Kalker - Haitsma - Robust video hashing paper Robust Signal Hashing may be used as a tool for video identification as well. An algorithm prepared by the Philips Research is an example for such purposes. Applications range from video monitoring on broadcast channels to filtering on peer- to-peer networks to meta-data restoration in large digital libraries. Proposed Video Hashing technique is 2-step algorithm that firstly extracts essential perceptual features from moving image sequences. Perceptual Images are computed in spatio-temporal domain from block-based DCT coefficients, which are based on simple statistics to avoid complex calculations. Secondly, the algorithm identifies any sufficiently long unknown video segment by efficiently matching the fingerprint of the short segment with a large database of pre-computed fingerprints. [14]

Category 2: Robust audio hashing prior study. Following studies are analyzed: Mihcak - Venkatesan : IHW01 - Robust Audio Hashing based on pseudo-random linear statistics in the time frequency domain. Burges - Platt - Jana : Distortion Discriminant Analysis. Özer - Sankur - Memon : Perceptual Audio Hashing. Haitsma- Kalker - Oostveen : Audio Hashing for Content Identification.

Robust Signal Hashing Algorithms may be used for identification of audio clips as well as database lookups in a way resistant to formatting and compression, similar to images and videos[4]. It is necessary to have accurate and efficient methods for searching, retrieving and classifying information from audio sources[15]. Robust Audio Hashing Algorithms are derived for these purposes. In a previous work on perceptual audio hashing algorithm by Mihcak[4], it is shown that while robust hash values are used primarily for general purposes, they may be used as improving other security concepts. For instance, these techniques can be used for watermarking robustness. Current watermarking methods use a secret key to generate and embed a watermark. However, if the same key is used to watermark different items, then each instance may leak partial information and it is possible that one may extract the whole secret from a collection of watermarked items. Thus it will be ideal to derive content dependent keys, using a perceptual hashing algorithm (with its own secret key) that is resistant to small

changes. In addition, this algorithm has randomness and unpredictability properties which are similar to cryptographic MACs. The techniques here are also useful for synchronizing in streams to find fixed locations against insertion and deletion attacks. Mihcak's algorithm has 4-stage methodology:

- The signal $\mathbf{X}$ is transformed to canonical form using Modulated Complex Lapped Transform. The result is the time-frequency representation of $\mathbf{X}$,
- Randomized interval transformation to time-frequency representation of $\mathbf{X}$ is applied in order to estimate audible statistics of the signal,
- Randomized rounding (i.e. quantization) is applied for secure dimension reduction
- Then, decoding stage of an error correction code is applied on quantized value of the statistics to map similar values to the same point.

In general there is a tradeoff between robustness and being informative, i.e., if very crude features are used, they are hard to change, but it is likely that collision may occur between hash values of perceptually different data. Robustness, in particular, is very hard to achieve. Mihcak "et al."[7] tries to solve this problem via applying highdimensional quantization.

In another work, Burges presents a linear, convolutional neural network with an oriented principal component analysis (OPCA) dimensional reduction[10]. Burges computes the hash values by finding linear projections which are as orthogonal as possible to the noise, but along which the variance of the original signal is concurrently maximized. He used OPCA so as to define the unit vectors defining the desired projections. The OPCA directions are defined as the directions $\mathbf{n}$ that maximize the generalized Rayleigh quotient [16]

$$\mathbf{q_o} = \frac{\mathbf{n}'C_1\mathbf{n}}{\mathbf{n}'C_2\mathbf{n}} \tag{2.4}$$

Audio data (lectures, broadcast) give high dimensional characteristics for feature extractions, because they exist usually in long forms. In order to overcome such diffi-

culties, OPCA is applied in hierarchal way. For perceptually similar data issue and noise reduction, there is a 2-stage-preprocessing part. Distortion caused by frequency equalization, volume adjustment and distortions that cannot be heard by a human listener are removed respectively with these 2 parts: In the first preprocessing part; low-pass filter is applied in log spectrum. Then, the signal is uniformly lowered by 6 dB and clipped at 70 dB. The output of the first part is pair-wise difference between log spectrum of original signal and filtered, lowered, clipped signal. In the second preprocessing part; The log spectrum from the first step is exponentiated, and then a frequency-dependent perceptual threshold is generated. The output of the second part is the difference in dB between the log spectrum of original signal and the log perceptual threshold. After preprocessing part, the 2-stage-feature extraction part comes. These stages are in a way opposite, where first stage computes intermediate features by sliding a small window with an overlap by half(relatively big) over original data. While the second part computes the final (hash) values by sliding a large window with small overlap. This method, also called distortion discriminant analysis (DDA) –by the authors– is indeed one of the good algorithm for the stream audio fingerprinting task.

Most of the stated algorithms are not designed for general purposes. They have several special tasks where they are claimed to be better. In Philips Research Center, Haitsma proposes a novel algorithm for monitoring task in a particular database, which is computationally very efficient while leaving the "unique dimension reduction" behind. Its main strategy is to compute hash values from different frequency bands in real-time and search efficiently in a already prepared and arranged database. A distinguishing feature of the proposed hash scheme is taking human auditory system(HAS) into account and its ability to extract a bit string for every so many milliseconds[9]. Haitsma introduces two important facts. First, middle-length-audio files which are stored in a "moderately large database"(100000) can be identified with a sequence of 256 hash values, which approximately corresponds to 3 seconds of audio files. Unique identification is enough for this setup therefore bit error rates and possible collisions are not considered deeply for larger databases. Second, he stated that many important audio features live in the frequency domain. In order to keep these important au-

dio features (e.g. tones), spectral representation is computed by performing a Fourier transform on every frame. Sensitivity of the phase of the Fourier transform to different frame boundaries do not harm the structure because the HAS is relatively insensitive to phase. Therefore, only the absolute value of the spectrum is retained. The proposed algorithm is as follows: 33 non-overlapping frequency bands ranging from 300Hz to 3000Hz are selected. These bands have a logarithmic spacing which is chosen, because of the HAS also operates on approximately logarithmic bands (the so-called Bark scale). The algorithm bases on the sign of spectral energy differences, which is proved to be very robust to many kinds of processing. The bits of the robust hash string are formally defined as:

$$H(n,m) = \begin{cases} 1 & \text{if} \quad EB(n,m) - EB(n,m+1) \\ & \qquad - (EB(n-1,m) - EB(n-1,m+1)) > 0 \\ 0 & \text{if} \quad EB(n,m) - EB(n,m+1) \\ & \qquad - (EB(n-1,m) - EB(n-1,m+1)) \leq 0 \end{cases} \tag{2.5}$$

Here, m stands for frequency band, n stands for frame, *EB()* stands for energy band. *H(n,m)* is the output m-th hash bit of n-th frame. After forming the hash vector, an efficient search algorithm is applied for database look-up. Audio signals vary radically in time domain and (as it is already explained) many important audio features live in the frequency domain, consequently robust audio hashing is generally studied in this domain. Also every audio has a short-time inherent periodicity. Summing these facts with HAS give birth to the mel-frequency cepstrum(MFC) which is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency[17]. In Bogazici University, a study on robust audio hashing take into consideration of spectral properties, short term periodicity and MFC coefficients. Frame-by-frame fundamental period and the singular value decomposition of the Mel frequency cepstral parameters

are the basis of this work. Two perceptual audio hashing functions are investigated. One of them operates in the time domain, and uses the periodicity of audio signals. The time characteristics of the dominant frequencies of the audio track consists of the discriminating information. In order to measure the periodicity, two different approaches are used, which are estimation-based and correlation based techniques. A least-square periodicity estimator (LSPE) is applied to compute the period value of each frame. Correlation based technique is simply peak picking in the correlation sequence of the preprocessed audio signal. Second one uses time-frequency domain, which is driven by the frame-by-frame MFCC coefficients. Dimensional reduction is provided by singular value decomposition of computed time-frequency domain matrix.

Audio Signal is segmented into F frames and 13 MFCCs are computed as well as their first and second derivatives, concluding 39 MFCCs (we denote it by M). As a result an FxM dimensional MFCC-feature matrix that represents the audio signal is formed. Singular value decomposition (SVD) reduces the FxM-dimensional MFCC-feature matrix into a much smaller invertible square matrix. Thus, the given FxM matrix is decomposed as $A = UDV^T$, where A is the FxM matrix that we want to summarize, D is an FxM matrix with only min(FxM) diagonal elements, U is an FxF orthogonal matrix, and V is an MxM orthogonal matrix. In general, a few singular values (first few components of the diagonal matrix D give a good summarization of the matrix A [11].

There are other uses of Robust Signal Hashing other than anti-piracy search.

- Mihcak - Venkatesan - Tie Liu: Robust Image Watermarking based on Quantization of Pseudo Random Linear Statistics. A watermarking algorithm may be counted as this kind of study. Mıhçak introduces a “robust image watermarking” approach based on quantization of pseudo random linear statistics. By using wavelets domain, robust semi-global statistics of images are derived, randomized and quantized in order to embed the watermark. An essential emphasis (concerned)of the proposed method is randomization, which is crucial for security and robustness against arbitrary quality-preserving attacks.[18] In addition, Can-

nons employs hashing as an aid to watermarking. He used robust signal hashing as side information in a semi-blind watermarking system. where secret subset of the full-frame discrete cosine transform of an image is computed[19].

- Harmanci - Mihcak: They used robust image hash values in order to achieve temporal synchronization for a video watermarking application

2. PROPOSED SPEECH HASHING APPROACH

2.1. Notation

Boldface letters represent vectors; the corresponding regular letters together with explicitly-specified indices denote individual elements. For instance, for a vector $\mathbf{a}$, $a(i)$ represents the i-th coefficient of $\mathbf{a}$.

2.2. Proposed Speech Hashing Algorithm

Although the problem looks awfully similar to robust audio hashing, one of the fundamental goals of this work is to utilize the fact that the underlying signal is indeed a speech signal (which has radically different characteristics from an audio signal). Contrary to English, out-of-vocabulary (OOV) rates are considerably high for Turkish language. Thus, additional methods are required to eliminate the ill effects of OOV queries and inaccurate automatic speech recognition (ASR) on spoken term detection [20]. We can extend this inference for speech data tracking, which is our primary aim. Instead of using Hidden Markov Models (HMM), we use phone based, frequency-time domain analysis for developing a fingerprint for any speech data. For track and trace applications, where HMM's may not be feasible and may be impossible to build, several approaches are considered for audio [4][9][11]. Most analyze the general frequency or time domain characteristics. None has ever considered any inside information that lies in the speech data. The information that is special to speech may be phonetic, verbal or linguistic. In this work, phonetic characteristics are chosen for hash calculation. We take phones as speech characteristics and randomized frequency as hashing backbone so as to conclude on a secure speech tracking.

- The outline of the proposed algorithmic approach is as follows:
    1. *(Offline Stage)* Extract most effective (most frequent) letter (vowel) patterns in the cepstral domain. We may see this stage as a training part. From different test subjects, letter patterns are taken and average letter pattern

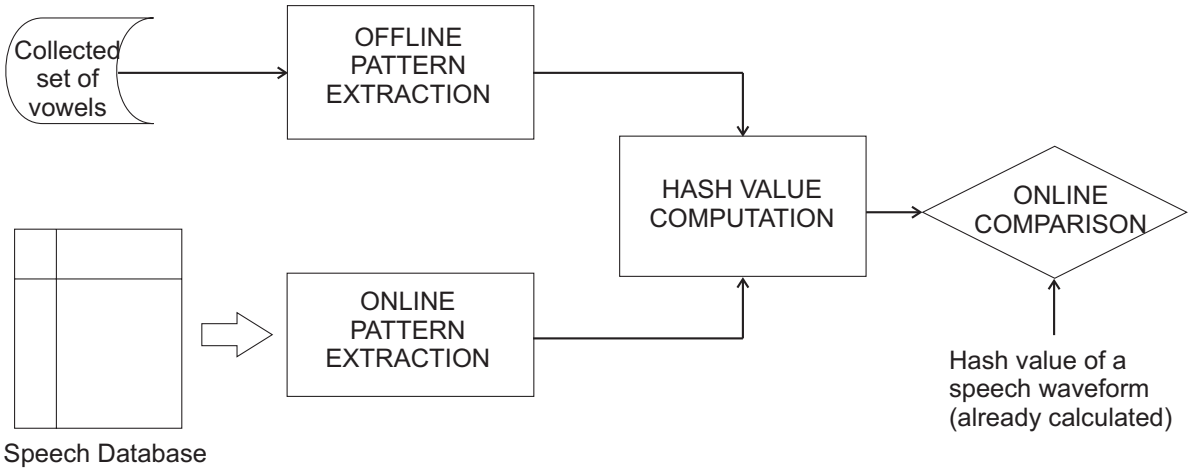

Figure 2.1. Proposed Speech Hashing Algorithm Overall Flow.

is prepared.

2. *(Online Stage - Hash Extraction)* For each input speech, of which hash value needs to be found, do the following:
   (a) Transform the input speech signal to the time-frequency domain using Short Time Fourier Transform (STFT). Speech signals have short-time inherent periodicity therefore STFT is used.
   (b) Derive the cepstrum domain coefficients of the input speech signal
   (c) For each time segment, project the cepstrum coefficient on the subspace spanned by the pattern that represents the letter (vowel) at hand.
   (d) Apply a pseudo-random linear transformation to the aforementioned projection outputs, such that each element of the output vector of this transformation corresponds to a weighted linear combination of the projection outputs that were found at the end of the previous stage and the weights are generated via using a secret key as the seed of a secure PRNG (pseudo random number generator)
   (e) A "fine-quantized" version of the transformation output of the previous stage constitutes the robust hash value of the speech signal of interest
3. *(Online Stage - Comparison)* We say that two speech signals of interest are "perceptually similar" (resp. "perceptually different") if their robust hash values are sufficiently close (resp. far apart) in the sense of Euclidean norm. In practice, we apply a simple thresholding to the L_2 distance between the

hash values of the two speech signals that are tested.

2.2.1. Offline Pattern Extraction

Feature extraction is the first step of any hashing functions. In order to prepare robust, efficient and distinctive hash functions, feature extraction must satisfy four condition.

1. (As far as hashing is concerned) Dimensionality reduction is the first fundamental concept that a feature extraction mechanism must satisfy. Randomized frequencies of similar tiny signals are analyzed in this work.
2. Robustness is another fundamental concept. The features must be robust to several distortion, because our topic is robust signal hashing. Time-frequency domain features are used which are more robust than time features.
3. The calculated features must be informative, where we are using speech characteristics (explicitly vowels) to achieve this goal better.
4. The feature extraction operation must be computationally efficient[10]. The computational efficiency is still being studied and several approaches are presented as a future work.

Similarity in the spectral domain is our starting point where we check the similarity by patterns. Therefore there should be a prominent and a distinctive pattern. In our work, we propose phonemes(vowels) as the distinctive pattern. Features are extracted from the phonemes.

Phonemes may be seen as the smallest unit in the sound system of a language for speech processing. They are minimal units that serve to distinguish between meanings of words[21]. Spectral domain characteristics(more specifically -spectral amplitude-) are the most important and simplest facts to discriminate the phonemes. By analyzing the amplitude changes in spectral domain a phoneme can be studied in a single or group of three phones, which are called monphone and triphone respectively. Monophone is the individual units of sound that make up a word. Triphone is the elementary

linguistic unit that represents a sequence of three phonemes. Here we come with a novel combined perspective. We consider the sounds that make up the word as individual-monophone- while using the middle segment of sequence of three phonemes-triphone-. Middle segment is used so that the errors by interaction between phones are minimized to come up with more robust features. Generally speaking, Turkish has a nice property about phonemes. We may consider each letter as a phoneme. From this point of view, we consider a vowel's frequency domain characteristics as our pattern.

As we use pattern of phonemes for our features, the selected phonemes become important. There is a preliminary stage, where we find most frequent vowels in Turkish. There is a statistical study on online written broadcast and story databases for this purpose. To start with, we found 'a' and 'e' as the most frequent two vowels in Turkish.

In the Offline Pattern Extraction part, after determining the most frequent Turkish phonemes and their patterns, from different test subjects, different files, vowel patterns are taken and average letter pattern is prepared. In order to generalize the robust hashing function, the distinctive pattern should be as general as possible. Also the considered pattern may be distorted locally, some phonemes may be unnaturally loud or too quiet or there may be variations in energy in the collected patterns, which bring out difficulty to form basic spectral pattern. In addition, there may be several noise sources such as impulsive noise that can not be estimated. In order to reach a uniform and more naturally sounding loudness, these factors must be reduced. Therefore we apply amplitude normalization.

Amplitude normalization is a very effective procedure to filter out variations and non-specific noise. The ratio between the maximum and the minimum of the energy is decreased and a more smooth waveform is obtained.

We compute amplitude normalization via applying the following formula (formula is applied window by window like a low pass filter):

$$NormalizedInput = \frac{(InputData(:) - min(InputData))}{(max(InputData) - min(InputData))} \tag{2.1}$$

Human beings are most sensitive to a certain group of frequencies. Average speech spectrum for each sex is between 300 Hz and 4K Hz. (the most relevant frequency range). Ideally as far as speech waveform is concerned the hash function should be convenient to that range. Lower and higher frequency components are unnecessary to characterize speech. After collecting the spectral patterns and forming a basic representation of a phoneme, the redundant frequencies are discarded via a band pass filter.

In a nutshell, in the offline phase, the *spectral patterns* of the Turkish letters, which will be used to derive the actual robust speech hash during the online phase, are computed. Such patterns can be perceived as representations of chosen Turkish letters in the spectral (Fourier) domain. Once such patterns are extracted, then they are used as a set of filters which will be applied to an input speech signal to derive the desired spectral features via simple FIR (finite impulse response) filtering (which can also be perceived as finding out the correlation values between the spectral pattern and the input signal coefficients in the spectral domain). As it turns out, these features form a "time series" for any speech signal. Once this step is completed, a *pseudo-random linear transform* is applied to the outcome, thereby producing the proposed robust hash vector. At this stage, a secret key is used as the seed of the pseudo-random number generator to achieve the desired security level.

A step-by-step algorithmic description of the proposed technique is given below.

Offline Phase :

We assume that we are given a total of L training signals, $\{\mathbf{s}_i\}_{1 \leq i \leq L}$ and we are running our algorithm for a specific Turkish letter of choice, α. Furthermore, we also assume

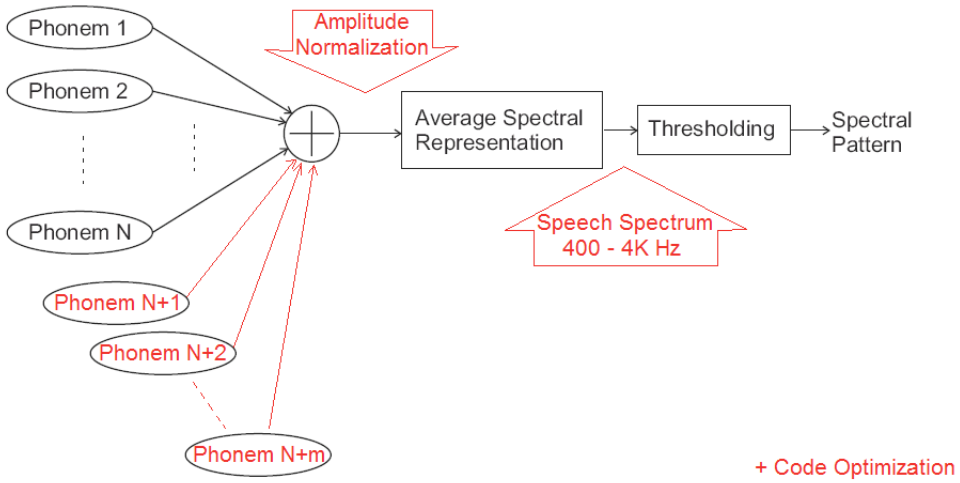

Figure 2.2. Offline Phase: Spectral Pattern Extraction

that a well-accepted and high-performance ASR (automatic speech recognition) engine has been applied on these training signals, thereby producing the letter locations using a tri-phone model.

1. Given a letter α, and a set of L training signals $\{\mathbf{s}_i\}_{1\leq i\leq L}$ (together with their ASR outputs), we "collect" the initial vectors (patterns) $\left\{\tilde{\mathbf{p}}^{\alpha}_{j,i}\right\}$ that represent the central part of the tri-phone representation, where $1 \leq i \leq L$, $1 \leq j \leq M^{\alpha}_i$; L is the number of training signals and M^{α}_i is the number of occurrences of the letter α within $\mathbf{s}_i$. Amplitude Normalization is applied in this stage.
2. We compute the length-N_{FFT} of each initial pattern $\tilde{\mathbf{p}}^{\alpha}_{j,i}$ and find out the magnitude of each FFT coefficient, yielding the "Fourier-magnitude" vector $\tilde{\mathbf{q}}^{\alpha}_{j,i}$, where $1 \leq i \leq L$, $1 \leq j \leq M^{\alpha}_i$.
3. Next, we compute $\hat{\mathbf{q}}^{\alpha}$, the empirical mean of $\left\{\tilde{\mathbf{q}}^{\alpha}_{j,i}\right\}_{i,j}$; i.e.,

$$\hat{\mathbf{q}}^{\alpha} = \left(\sum_{i=1}^{L}\sum_{j=1}^{M^{\alpha}_i}\tilde{\mathbf{q}}^{\alpha}_{j,i}\right) / \left(\sum_{i=1}^{L} M^{\alpha}_i\right). \tag{2.2}$$

4. Next, we compute $\bar{\mathbf{q}}^{\alpha}$, a "$\beta$-percent frequency-filtered" version of $\hat{\mathbf{q}}^{\alpha}$ via dropping out least-magnitude Q coefficients of $\hat{\mathbf{q}}^{\alpha}$ where Q is calculated such that the L_2 norm of $\bar{\mathbf{q}}^{\alpha}$ is β-percent of the L_2 norm of $\hat{\mathbf{q}}^{\alpha}$. This task is carried out via solving

the following:

$$\text{sort}\left(\bar{q}^{\alpha}\right)(i) = \begin{cases} 0 & i \leq Q, \\ \text{sort}\left(\hat{q}^{\alpha}\right)(i) & \text{else} \end{cases} \tag{2.3}$$

where

$$Q \stackrel{\triangle}{=} \arg\min_{Q} \left| \frac{\sum_{i=1}^{Q} \text{sort}\left(\hat{q}^{\alpha}\right)(i)}{\sum_{i} \hat{q}^{\alpha}(i)} - (1-\beta) \right| \tag{2.4}$$

and sort $(\cdot)$ represents the sorting function in ascending order.

Remark: Note that; we propose to extract the data in the cepstral domain since years of research in speech processing showed us that data in cepstral domain is more reliable and more robust. We also observed as a result of our extensive experiments that the pattern data in the time domain are very much susceptible to minor local changes and shifts, where perceptual differences are difficult to find.

2.2.2. Online Hash Extraction

Robust Hash Values are found in this part. Pre-computed spectral patterns are used as our basis vectors. In the online hash extraction part, like in the offline, we apply amplitude normalization and speech spectrum. As stated above, a speech file may be distorted locally, some phonemes may be unnaturally loud or too quiet or there may be variations in energy in the speech, which bring out difficulty to find out correct similarity between the waveform and basis spectral pattern computed in the offline stage. In order to reach a uniform and more naturally sounding loudness, these factors must be reduced. We compute amplitude normalization via applying the same formula. In addition, we use the sensitiveness of humanbeings to a certain group of frequencies. Average speech spectrum for each sex is between 300 Hz and 4K Hz. Ideally as far as speech waveform is concerned the hash function should be convenient to that range. Lower and higher frequency components are unnecessary to characterize speech. The

redundant frequencies are discarded via a band pass filter, before comparison stage. Lastly, hash values are computed.

Online Phase :

In the online portion of the proposed algorithm, the vector $\bar{\mathbf{q}}^{\alpha}$ is used as the "spectral pattern" that represents the letter α as accurately as possible. During the online phase, the main idea is to find out the correlation between the spectral pattern of interest and the spectral representation of the input speech signal at each time segment, subsequently followed by deriving weighted linear combinations (with pseudo-randomly-generated smoothly-varying weights) of these correlation values over pseudo-random sufficiently-long time intervals. The resulting vector of linear combinations is termed as the "robust hash vector" of the input speech signal. A step-by-step algorithmic description of the online phase is presented next.

1. Given a length-N digital input speech signal $\mathbf{x}$, we partition it into non-overlapping N_{blocks}-many time blocks, i.e., $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_{blocks}}]$.
2. Amplitude Normalization is applied to whole signal, in order to get a uniform and an easier data to process.
3. Given an input letter α, the corresponding spectral pattern $\bar{\mathbf{q}}^{\alpha}$, and the input signal blocks $\{\mathbf{x}_i\}$, we calculate the resulting correlation vector $\mathbf{c}^{\alpha}$. In order to achieve this task, we do the following for each $i \in \{1, 2, \ldots, N_{blocks}\}$:
   (a) We compute the length-N_{FFT} of $\mathbf{x}_i$ and find out the magnitude of each FFT coefficient, yielding the "Fourier-magnitude" vector $\mathbf{u}_i$.
   (b) Next, we compute the correlation between $\mathbf{u}_i$ and $\bar{\mathbf{q}}^{\alpha}$, thereby forming $c^{\alpha}(i)$, the i-th element of $\mathbf{c}^{\alpha}$, via applying the following:

$$c^{\alpha}(i) = \frac{\sum_j u_i(j)\, \bar{q}^{\alpha}(j)}{\sqrt{\left(\sum_j [u_i(j)]^2\right) \cdot \left(\sum_j [\bar{q}^{\alpha}(j)]^2\right)}}$$

4. Using K as the seed of a secure PRNG (pseudo-random number generator), generate N' "regions" over time, denoted by $\{\mathcal{R}_j\}_{j=1}^{N'}$, such that, for each $j \in$

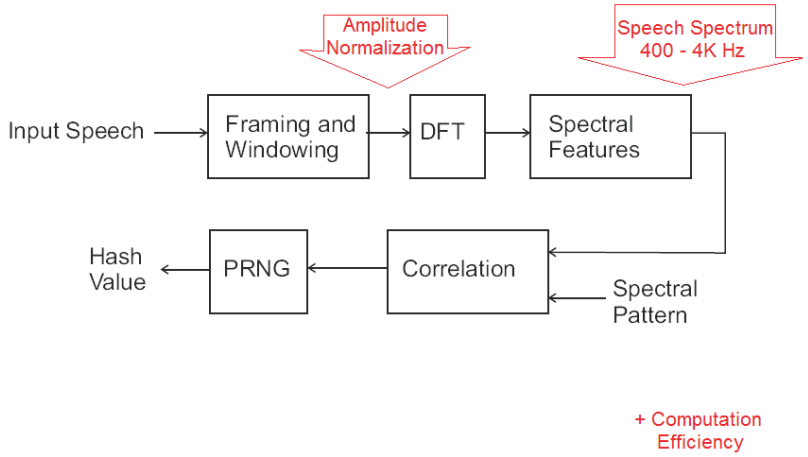

Figure 2.3. Online Phase: Hash value computation

$\{1, 2, \ldots, N'\}$, we have

$$\mathcal{R}_j = \{A_j, A_j + 1, \ldots, B_j - 1, B_j\} \subseteq \{1, 2, \ldots, N_{blocks}\},$$

where the length of each region, $L_j \stackrel{\triangle}{=} B_j - A_j + 1$ is uniformly distributed within the set

$$\{MinRegionSize, MinRegionSize + 1, \ldots$$
$$, MaxRegionSize - 1, MaxRegionSize\}$$

, and A_j is uniformly distributed within the set $\{1, 2, \ldots, N_{blocks} - L_j\}$ conditioned on L_j.

5. For each $1 \leq j \leq N'$, using K as the seed of a secure PRNG, generate the weight vector $\mathbf{w}_j \in \mathbb{R}^{L_j}$ such that $\mathbf{w}_j$ is a realization of a correlated Gaussian vector (with user-specified input parameters) such that the coefficients are zero-mean and they are smoothly-varying.
6. The output hash vector $\mathbf{h}_K^\alpha$ of the input signal $\mathbf{x}$ is then computed via using

$$h_K^\alpha(j) = \sum_{i=A_j}^{B_j} w_j(i - A_j + 1)\, c^\alpha(i), \quad \text{where} \quad 1 \leq j \leq N'.$$

2.2.3. Hash Based Perceptual Comparison

We simply use thresholding of the L_2 distance between the speech hash values of the two input speech signals of interest. In principle, it is also possible to potentially use other types of "classification" methods that utilize these hash values. Such methods may potentially use standard machine learning techniques, such as Fisher Discriminant Analysis and/or Support Vector Machines. We leave such kinds of extensions for future research.

3. EXPERIMENTAL RESULTS AND DISCUSSION

We conducted several experiments in order to experimentally evaluate the performance of the proposed robust speech hashing algorithm. Here, since well-known audio hashing techniques can also be applied to the speech signals, the main approach is to compare the performance of our algorithm with such audio hashing methods. In order to achieve this task, we used the following audio hash techniques:

- Haitsma-Kalker-Oostveen Technique: (Haitsma) [9]
- Mihcak-Venkatesan Technique: (Mıhçak) [4]
- Ozer-Sankur-Memon-Anarim Technique: (Sankur) [11]

Nearly 80 samples of 10 speakers voices are used to train the background model of phonemes. After training the phoneme model, in order to evaluate the performance of the proposed algorithm and compare it with the existing techniques, the following attacks(pre-defined in CoolEdit software, therefore the attack names are directly cited) have been applied on a speech database. The speech database consists of over 5 hours of broadcast news programs: (VOA) and four different TV channels (CNN Turk, NTV, TRT1 and TRT2). Each speech waveform is about 7 minutes and 30 minutes long, and has been digitized with sampling frequency 16000 Hertz.

- AWGN: Additive White Gaussian Noise
- Back Ground Noise: Background noise applied between -60dB and -25dB
- Boost 10dB: Boost 10 dB applied to whole file
- Cut 10dB: Cut 10 dB from whole file
- EchoHeavy: Echo is applied uniformly
- midLevelExpansion3: Non linear random amplitude distortion
- midLevelExpansion4: Non linear random amplitude distortion -heavier than previous one-
- Notch Filter: attenuation around 15dB at 430 Hz, 1024 Hz, 1618 Hz, 2212 Hz, 2806 Hz, 3400Hz

- Old time radio: noise with attenuation at low frequencies and amplification at high frequencies
- Pitch Bending: Frequencies are raised or lowered, while keeping the tempo(time) of the original file

The aforementioned attacks have been implemented using Matlab and the *CoolEdit* software [22] which is commercially available.

We compared the proposed technique with the aforementioned audio hashing techniques in the presence of each of the aforementioned attacks in the sense of operational ROC (receiver operating characteristics). For each type of attack and for a specific detection threshold, we computed the empirical estimates of the probabilities of false alarm and miss, thereby forming the desired ROC results, which are presented next.

- Figures 3.1(a), 3.1(b) show the ROC curve in the presence of additive white gaussian noise. Here, as we see the proposed technique produces superior results with respect to Mıhçak's and Haistma's technique. Patterns are computed in spectral domain therefore proposed hash function is robust to amplitude distortions.
- Figures 3.6(a) , 3.6(b) , 3.7(a) and 3.7(b)show the ROC curve in the presence of midLevelExpansion attacks. Here, as we see the proposed technique produces superior results with respect to Mıhçak's and Haitsma's technique. Amplitude normalization and frequency based hash values make it robust to non-linear distortion attacks.
- Figures 3.2(a) , 3.2(b) , 3.3(a) , 3.3(b) , 3.4(a) and 3.4(b) show the ROC curve in the presence of several amplitude distortion attack. Here, as we see the proposed technique produces superior results with respect to Mıhçak's and Haitsma's technique. Our algorithm is superior to other audio hash algorithms, especially considering amplitude attacks. Spectral domain based hash value computation is carried out via random rectangles. In addition pre-work amplitude distortion gives an easier data to process.
- Figures 3.8(a) , 3.8(b) , 3.10(a) , 3.10(b) , 3.12(a) , 3.12(b) and 3.13(b) show the

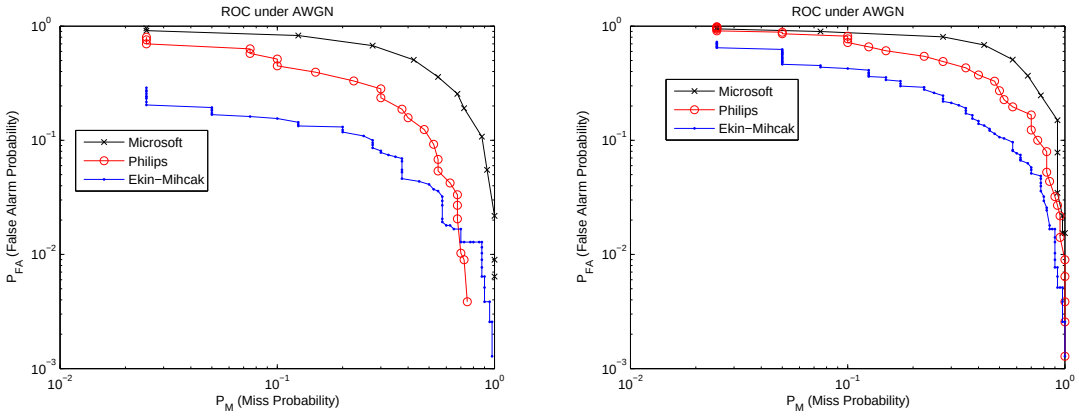

Figure 3.1. ROC under Additive White Gaussian Noise.

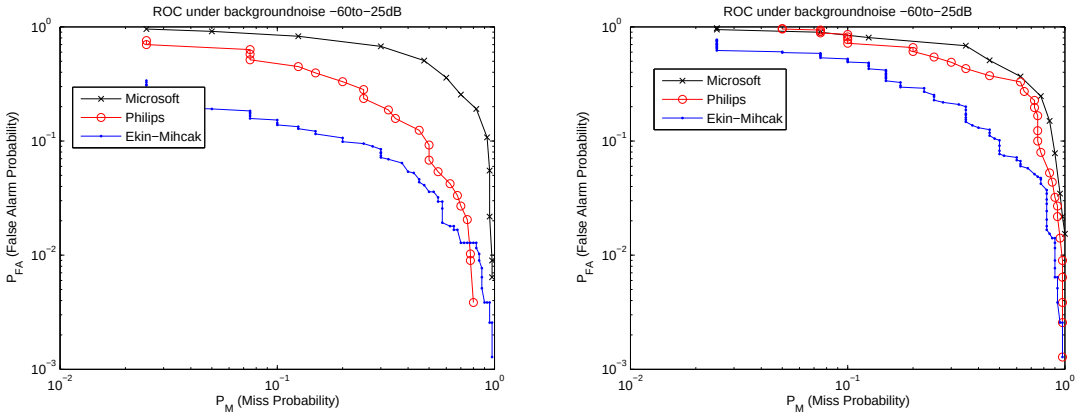

Figure 3.2. ROC under Back Ground Noise.

ROC curve in the presence of several frequency selective distortion and directly spectral domain attack. Here, as we see the proposed technique mostly produces superior results with respect to other audio hash algorithms.

In the light of these results, we see that the proposed robust speech hashing technique demonstrated a "stable" behavior in the potential presence of "any" attack that preserves the perceptual fidelity of the input signal; that is, within a wide variety of attacks, the proposed technique produces "reasonable" results when compared to the state-of-the-art audio hashing techniques. On the other hand, these audio hashing techniques, which form a benchmark for us, have been observed to produce good results for some specific classes of attacks and more moderate results for other classes of attacks. In the final version of our proposed technique, we believe that it will surpass most of

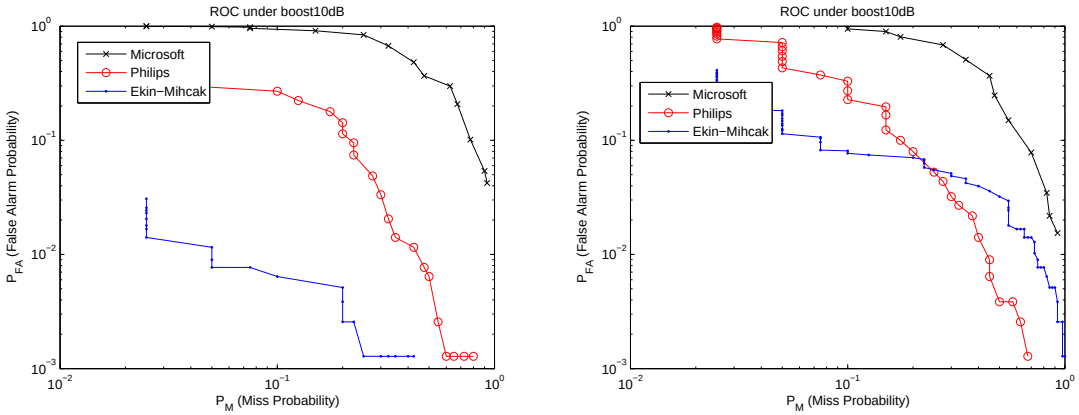

Figure 3.3. ROC under 10dB boost attack .

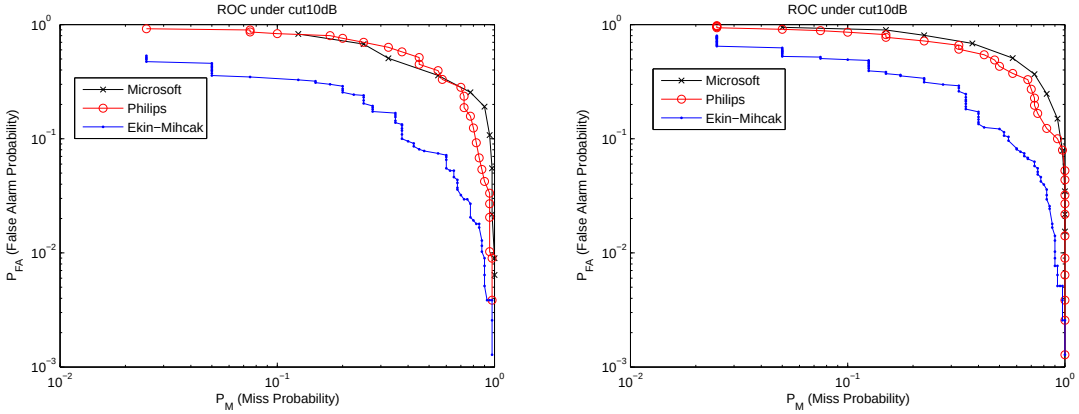

Figure 3.4. ROC under 10dB cut.

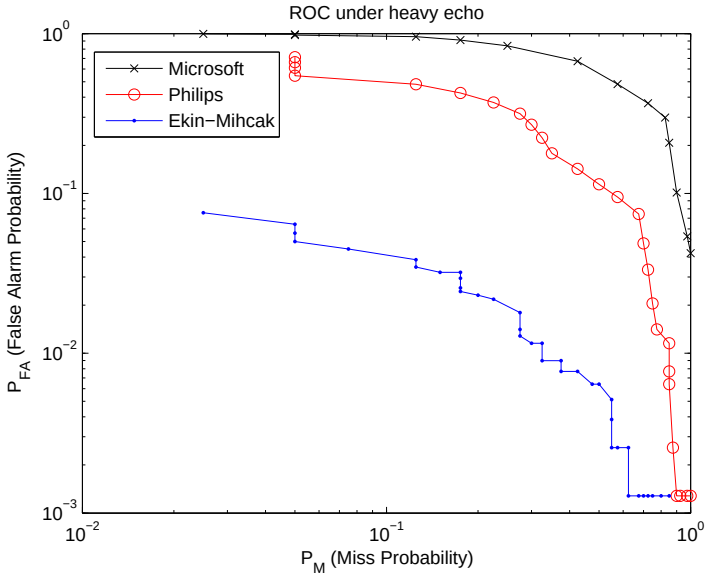

Figure 3.5. ROC under heavy echo attack.

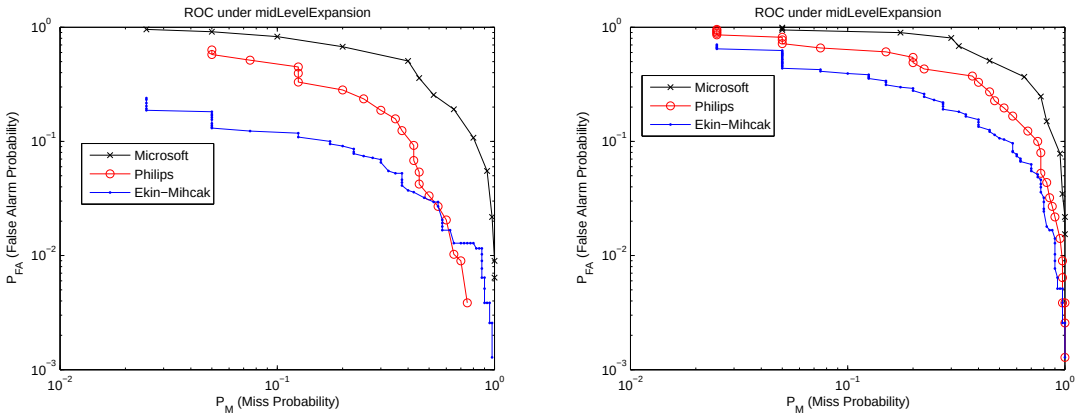

Figure 3.6. ROC under non linear distortion attack.

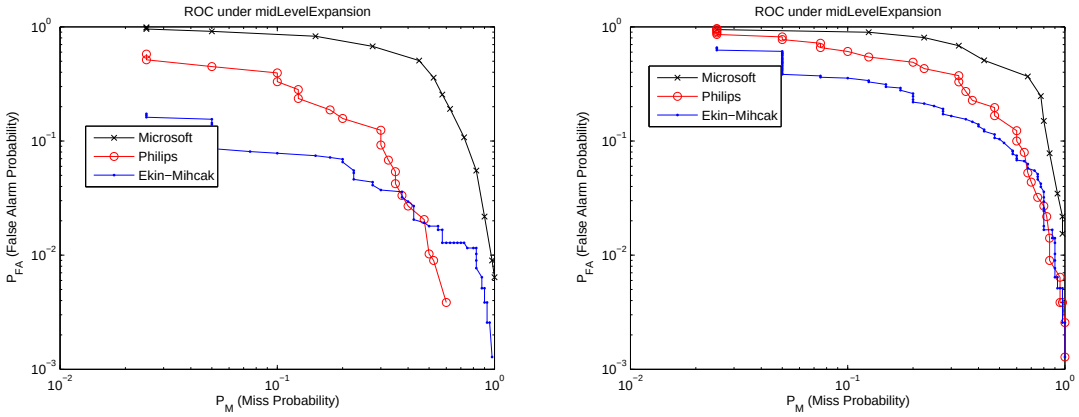

Figure 3.7. ROC under heavy non linear distortion attack.

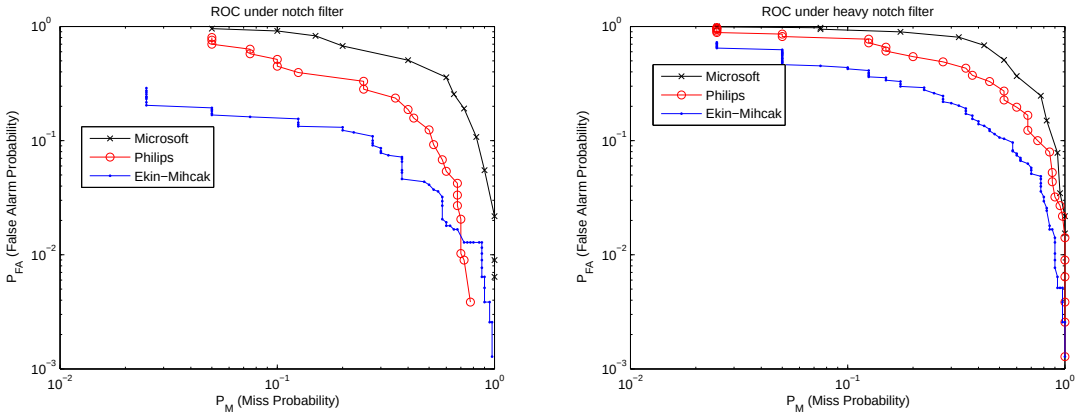

Figure 3.8. ROC under heavy notch filter attack.

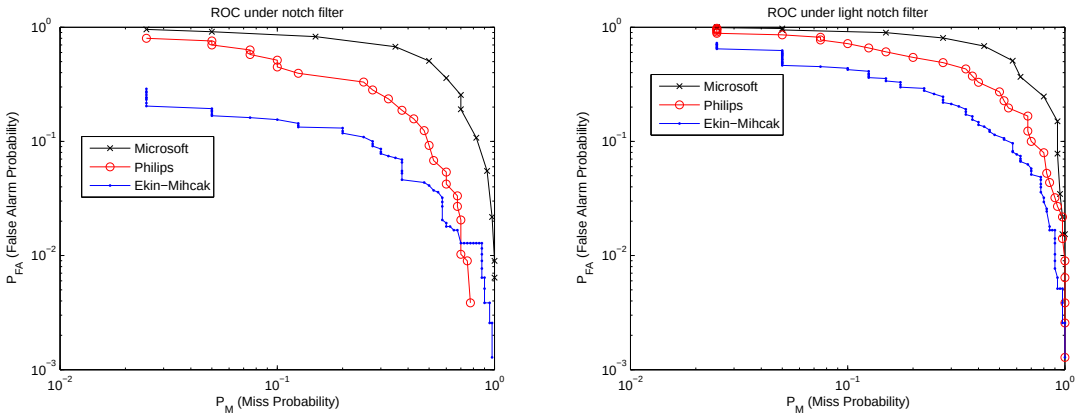

Figure 3.9. ROC under light notch filter attack.

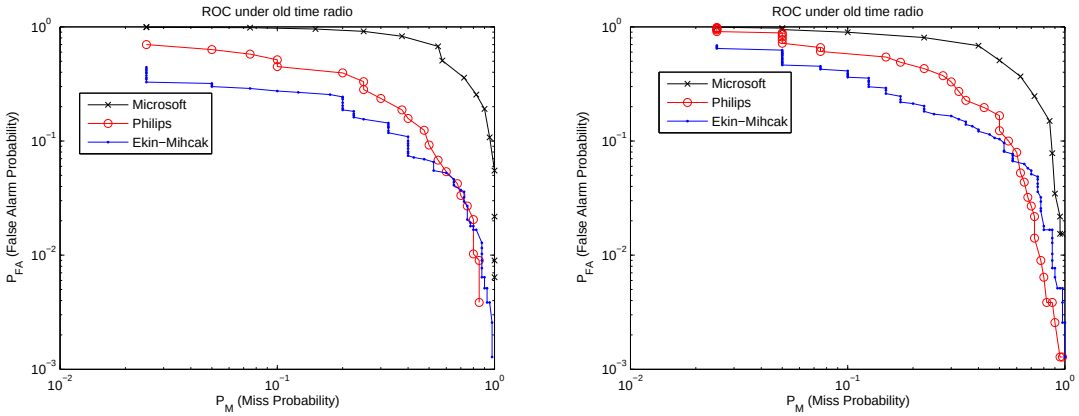

Figure 3.10. ROC under old time radio attack.

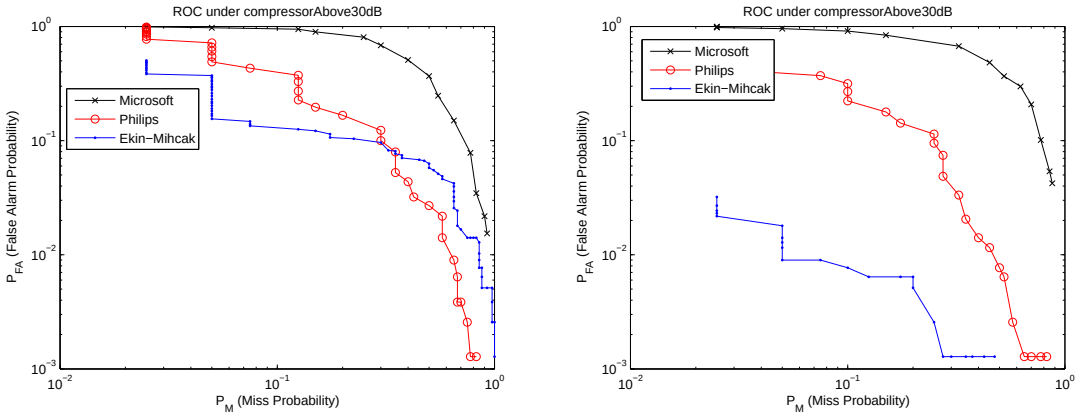

Figure 3.11. ROC under change in high energy values.

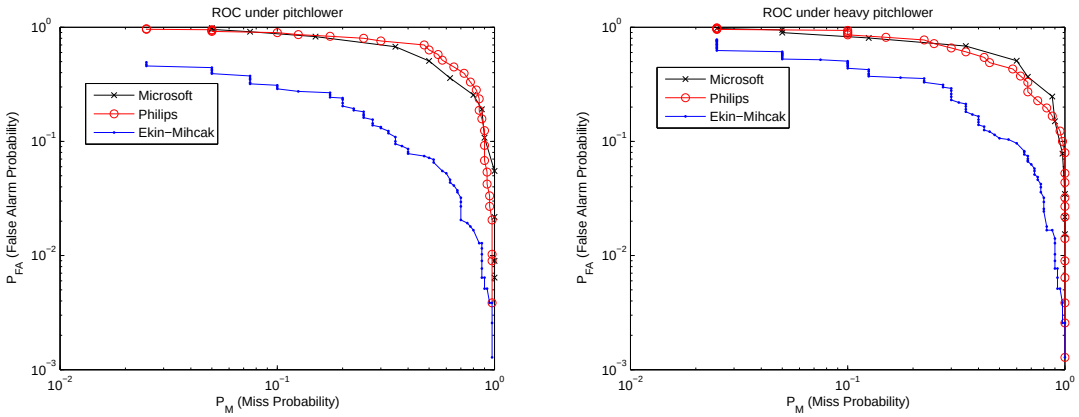

Figure 3.12. ROC under heavy pitch lower attack.

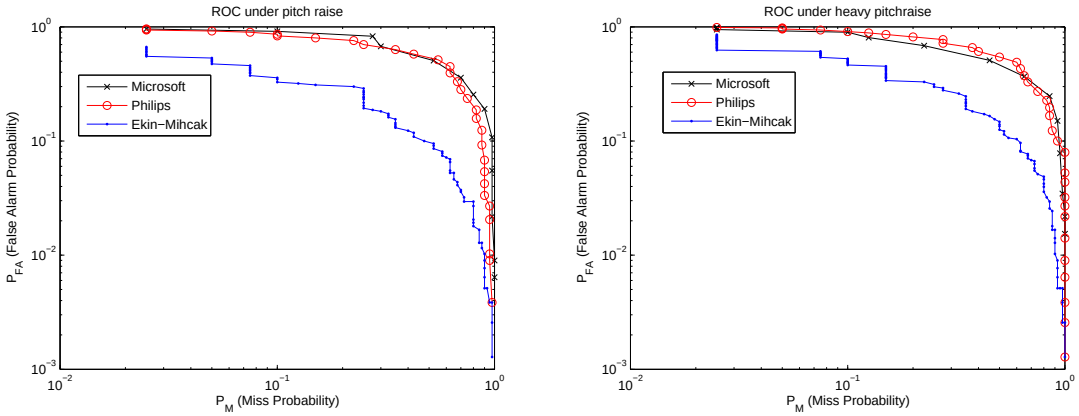

Figure 3.13. ROC under pitch raise.

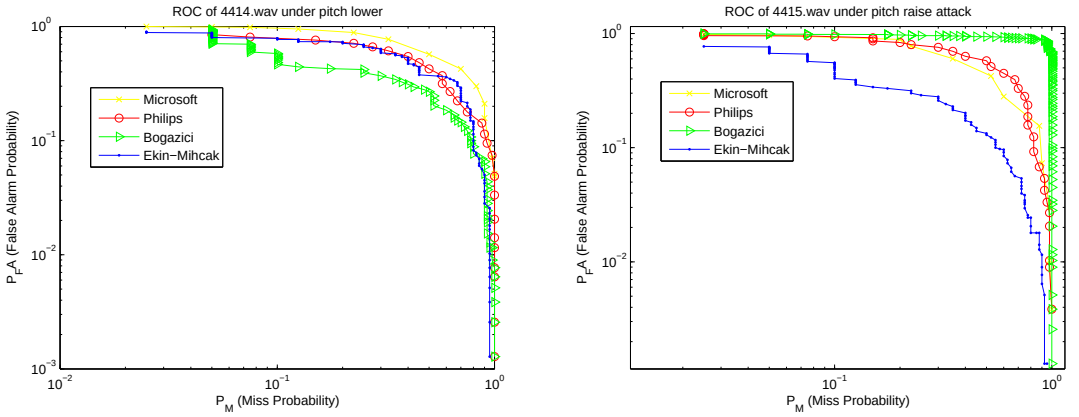

Figure 3.14. ROC under change in pitch of 2 different wave.

the existing audio hashing techniques, simply because we utilize signal characteristics that are specific to speech signals and do not apply to audio signals. For instance, we exploit HMM-based ASR outputs during the offline phase of our algorithm.

3.1. Potential Developments

The proposed hashing technique admits several input parameters that provide a design flexibility. As such, it is necessary to "experimentally optimize" these parameters in order to get good results in the potential presence of a wide variety of results. This task will be carried out during the rest of our research study. We will also review the utilized implementations of the existing robust audio hashing techniques and correct any potentially-existing bugs.

The experimental results that are provided in Sec. 3 are generated using the currently available software implementations of the existing audio hashing techniques. However, in order to make a fair comparison, it is necessary to adjust the parameters of all systems such that the lengths of the resulting hash vectors are the same, which constitutes another portion of the tasks that will be carried out during the rest of our research study. In addition to the existing audio hash techniques, the audio hash algorithm by Burgess et.al. [10] will also be included in the baseline study during the rest of our research study so as to augment our experimental results.

We plan to augment the performance of the proposed robust speech hashing algorithm, for instance, via applying appropriate techniques from machine learning literature. In particular, we plan to utilize PCA (principle component analysis) or OPCA (oriented PCA) approaches to extract the spectral pattern $\bar{\mathbf{q}}^{\alpha}$ during the offline phase of our algorithm. Another potential improvement might be due to the potential usage of the cepstral domain instead of the magnitude-Fourier domain in the extraction of the $\bar{\mathbf{q}}^{\alpha}$. Currently, the results have been obtained via using a single letter from the Turkish alphabet (in particular, the letter a). In the final version of the algorithm, we intend to use several other letters from the Turkish alphabet which would potentially be beneficial in terms of capturing the inherent speech-specific properties of the input

signals.

4. CONCLUSION

In this work, "Robust Speech Hashing" algorithm is proposed. We use Short Time Fourier Transform (STFT), because speech signals have short-time inherent periodicity. The cepstrum domain coefficients of the input speech signal are derived. Also thip part will be improved(we leave this work as future work). Hashing property of secrecy is satisfied via applying a pseudo-random linear transformation to the aforementioned projection outputs, such that each element of the output vector of this transformation corresponds to a weighted linear combination of the projection outputs. The weights are generated via using a secret key as the seed of a secure PRNG. A "fine-quantized" version of the transformation output of the previous stage constitutes the robust hash value of the speech signal of interest

We use over 1 hour speech from Voice over America and TRT broadcast database. Several attacks have been applied varying from additive white noise to frequency changing. The results are inspiring that proposed algorithm overwhelms most of the outcomes of the robust audio hashing techniques. Especially robustness to amplitude distortions and small frequency changes are provided. Robust Signal Hashing is designed for information security application such as leakage tracking. If we evaluate our system in terms of its performance ROC-wise, we may conclude that it is better with respect to robust audio hashing techniques that are considered. On the contrary, several drawbacks are reached such as computation complexity. Although one of our primary aim is to offer a computationally efficient algorithm, process details result a slow-down in the computation. The projection and windowing parts should be improved (such as applying different transform and pattern matching algorithms).

With the motivation of exploring the robust signal hashing area via applying speech characteristics, several add-on's are studied to extend robust audio hashing functions. Using vowels as phonemes and spectral analysis is our starting point while $L2$ norm is the comparison notion.

In addition, we will give weight to improving computation efficiency. Instead of directly comparing, there may be faster computation in spectral domain without windowing. Machine learning is another concept that may offer improvement in our system. Also, by implementing different vowels in pattern extraction part and if several different hash functions are computed for different vowel patterns, somehow we may merge these and generate better hash values.

To sum up, "robust speech hashing" problem is defined and studied. Although several signal types are considered like image and audio, there was no explicit study for speech signals on this problem. All in all, the dominant work is determining and assigning speech characteristics in order to come up with a novel algorithm.

REFERENCES

1. Schmucker, M. and P. Ebinger, "Promotional and Commercial Content Distribution based on a Legal and Trusted P2P Framework", *Procs of the 7th International Conference on E-Commerce Technology (CEC'05)*.

2. Troullinos, G., "A software based approach to secure voice applications", *Proceedings of the Third IEEE International Conference on Electronics, Circuits, and Systems, 1996. ICECS '96.*, Vol. 1, No. 1, pp. 176 – 182, Oct 1996.

3. Boneh, J., D.; Shaw, "Collusion-secure fingerprinting for digital data", *IEEE Trans. on Information Theory*, Vol. 44, No. 5, pp. 1897 – 1905, Sept 1998, `00705568.pdf`.

4. Mıhçak, M. K. and R. Venkatesan, "A Perceptual Audio Hashing Algorithm: A Tool for Robust Audio Identification and Information Hiding", *Procs. of the 4th Information Hiding Workshop*, Vol. 2137 of *Lecture Notes in Computer Science*, pp. 51–65, Springer, Pittsburgh, USA, April 2001.

5. Monga, V. and K. Mıhçak, "Robust Image Hashing Via Non-Negative Matrix Factorizations", *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 2, pp. 14–19, May 2006.

6. Sutcu, Y., H. T. Sencar, and N. Memon, "A secure biometric authentication scheme based on robust hashing", *Procs. of the 7th workshop on Multimedia and security*, pp. 111–116, New York, USA, October 2005.

7. Mıhçak, M. K. and R. Venkatesan, "New Iterative Geometric Methods for Robust Perceptual Image Hashing", *Procs. of ACM Workshop on Security and Privacy in Digital Rights Management*, Philadelphia, USA, 2001.

8. Venkatesan, R., S. Koon, M. Jakubowski, and P. Moulin, "Robust Image Hashing", *Procs. of the IEEE International Conference on Image Processing*, Vancouver,

Canada, 2000.

9. Haitsma, J., T. Kalker, and J. Oostveen, "Robust Audio Hashing for Content Identification", *Procs. of the International Workshop on Content-Based Multimedia Indexing*, pp. 117–125, Brescia, Italy, September 2001.

10. Burges, C., J. Platt, and S. Jana, "Distortion Discriminant Analysis for Audio Fingerprinting", *IEEE Transactions on Speech and Signal Processing*, Brescia, Italy, October 2001.

11. Özer, H., B. Sankur, N. Memon, and E. Anarım, "Perceptual Audio Hashing Functions", *EURASIP Journal on Applied Signal Processing*, , No. 12, pp. 1780–1793, 2005.

12. Kozat, S. S., R. Venkatesan, and M. K. Mıhçak, "Robust Perceptual Image Hashing via Matrix Invariants", October 2004.

13. Sun, R. S. H. and T. Yao, "An SVD and quantization based semi-fragile watermarking technique for image authmtication", *6. Annual Conference on Signal Processing*, Vol. 2, pp. 1592–1595, August 2002.

14. Oostveen, J., T. Kalker, and J. Haitsma, "Visual Hashing of Digital Video: applications and techniques", *Procs. of SPIE Applications of Digital Image Processing XXIV*, San Diego, USA, October 2001.

15. Johnson, S. E. and P. C. Woodland, "A Method for Direct Audio Search with Applications to Indexing and Retrieval", *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, Vol. 3, pp. 1427–1430, Istanbul, Turkey, 2000.

16. Diamantaras, K. and S. Kung, *Principal Component Neural Networks*, New York: Wiley, 1996.

17. Stevens, S. S., J. Volkmann, and E. B. Newman, "A Scale for the Measurement

of the Psychological Magnitude Pitch", *The Journal of the Acoustical Society of America*, pp. 185–190, 1937.

18. Mıhçak, M. K., R. Venkatesan, and T. Liu, "Watermarking via optimization algorithms for quantizing randomized semi-global image statistics", *Multimedia Systems*, Vol. 11, No. 2, pp. 185 – 200, December 2005.

19. Cannons, J. and P. Moulin, "Design and statistical analysis of a hash-aided image watermarking system", *IEEE Transactions on Image Processing*, Vol. 13, No. 8124714, pp. 1393 – 1408, October 2004.

20. S. Parlak, M. S., "Spoken Term Detection for Turkish Broadcast News", 33^{rd} *International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, pp. 5244–5247, Las Vegas, USA, April 2008.

21. Burquest, D. A. and D. L. Payne, "Phonological analysis: A functional approach", *SIL International*, p. 314, 1993.

22. [Online], Available: http://www.syntrillium.com/cooledit.