Music Streaming Service with Audio Recognition and Steganography Features

Adithya Natarajan[1], BM Amitraj[2], Cleva Vanessa Pinto[3], GVS Praneeth[4], Pallavi R[5]

[1-4]UG Student, Department of Computer Science and Engineering, Sir M. Visvesvaraya Institute of Technology, Bangalore, Karnataka, India
[5]Associate Professor, Department of Computer Science and Engineering, Sir M. Visvesvaraya Institute of Technology, Bangalore, Karnataka, India

---***---

Abstract - *The rapid spread in digital data usage in many real-life applications in the last decade has paved the way for the creation of several solutions to improve the convenience of consuming data, and has also urged new and effective ways to ensure their security. Amongst several technologies that have started to gain widespread use recently, Streaming is arguably one of the most popular. The premise of streaming is to provide access to data instantly, which essentially eliminates the need to store things locally on one's device. Music Identification has also seen a huge increase in popularity in the past decade from its use in some commercial applications and in copyright enforcement systems. A new kind of secret communication technology called Audio Steganography is by embedding a message into an existing file, like a music file, which can then be inconspicuously sent across the internet. The proposed application incorporates all these features to create an all-encompassing interface where the users can get access to music on-demand, identify unknown tracks within just a few seconds and can also share songs with secret messages embedded in them. To further enhance the user's experience, the application includes a collaborative filtering-based music recommendation system, a playlist creation feature, and a messaging system which facilitates the use of the steganography system.*

Key Words: streaming, recommendations, fingerprinting, steganography, messaging

1.INTRODUCTION

The Music Industry has been through several changes in the past decade thanks to the advancement of computer technology. From the way in which artist record and distribute their music, to the way in which consumers listen to music, everything has been impacted by computer technology.

From the days where the only way to listen to music was to watch it live, to the popularization of vinyl records, cassettes and CDs, to the current trend of streaming music, the way in which we listen to music has changed dramatically. Streaming platforms have made the distribution of music very efficient and fans can now very easily listen to the newest music from all their favourite artists, as soon as they are released. Fans can also create their own playlists, which may be seen as albums of their own, in which they can store the songs that they desire.

Suppose you hear a song in a local café that you like, the ability to be able to identify that song and save it to your library in just a few clicks is possible using modern Music Identification systems. The ability of these systems to identify the music in a noisy environment in an instant is achieved by technologies like Audio Fingerprinting and is used in popular applications like Shazam and also finds use in Copyright Enforcement Systems, like on YouTube.

Another feature which is commonly present in the above systems is recommendations. Recommendations are unique to each user, and they are provided to each user based on the popularity of the song. Hence, more popular the song, more likely it is to feature in your recommendations list.

As the public internet continuous to be used to send and receive data, the methods of securing this data is constantly evolving. The technique known as Steganography aims to conceal data within other types of media in such a way that its presence cannot be detected in any way to any unwanted recipient without doing any visible change in the cover media.

The aim of the proposed application is to integrate the features of Music Streaming, Music Identification and a message encryption technology known as Audio Steganography into a single, easy to use web interface so that users can access it from any device. By integrating this suite of services into a single application, the goal is to demonstrate how these technologies can work together to provide an all-around complete Music based service to cater to the needs of various users.

2. LITERATURE SURVEY

The Literature Survey is divided into 3 sub-sections, each detailing the existing reach in the field of Music Streaming, Audio Fingerprinting and Audio Steganography.

2.1 Music Streaming

The on-demand music feature of the proposed service is based on the Client-Server Music Streaming technology which delivers the contents of a music file in a gradual and continuous stream of information from a remote server to a user's computer.

The paper by G. Kreitz *et al* [1] details the working of the Music Streaming Service "Spotify". The paper provides an overview to the Spotify architecture and also explains some of the client-server protocols that the system uses. The Spotify system uses a traditional Client-Server architecture and alongside, uses peer-to-peer protocols to offload the server. While our system uses only a Client-Server architecture, the paper still explains various technical details like the protocols that the system uses, and their advantages. Notably, the system uses Transmission Control Protocol (TCP) rather than using User Datagram Protocol (UDP) like traditional streaming applications. This is due to the need for a reliable transport protocol and also due to the TCP congestion control system being suitable.

Dapeng Wu *et al* [2] explains the current approaches and challenges related to streaming video over the internet. While some of the topics covered in this paper are not of relevance to our system, lots of research was done to evaluate the various protocols available to achieve streaming. The key area that we focus on is the protocol differences explained by the paper. TCP has advantages over UDP in that it supports congestion control, uses retransmission, and employs flow control. The disadvantage of using TCP however is the delays introduced by retransmission. This is the reason UDP is more commonly used in streaming applications.

2.2 Audio Fingerprinting

The Audio Identification System is based on the technology of Audio Fingerprinting. Audio fingerprint technology is the generic term for the technique of retrieving an audio signal identical to that given as the query (the query signal) from a database (the stored signal) by comparing fingerprints, which are representations of the features extracted from these signals.

One of the most prominently referenced paper is by Wang A. L. [3] which details the approach used by the Shazam Audio Identification System. By using Spectrogram representation, peaks are extracted and represented as a Constellation Map. Fingerprints are derived using the technique of Fast Combinatorial Hashing, where a Target Zone is defined for each point in the constellation map (called the anchor point). The points in the target zone are paired with the Anchor, and each pairing is represented as a binary which encodes the frequencies and the time difference between the 2 points. These binary Fingerprints are then used as a key in a Lookup table along with the timestamp and the Song ID as the corresponding values. For matching a Query Song, after the fingerprints are generated, the matches are fetched from the database and the best match is determined by the song having the highest number of fingerprints with similar relative offsets from the start of the song.

An improvement to the Shazam Algorithm [3] was proposed by X. Sun *et al* [4] which suggested an alternate method of selecting Target Zones. Instead of using a Static Target Zone, this method proposed using a Dynamic Target Zone where, after the anchor point is selected, N peak points after anchor point are obtained as corresponding matching points. The advantage of this approach is that it avoids the target zone have having differing number of peak points. This greatly improves the matching accuracy when the query audio includes silent segments.

An alternate approach to the Shazam technique [3] was proposed by Jianhua Shi *et al* [5], whose algorithm used a different method of Feature Extraction. Here, the salient points are defined as points in the spectral representation which are local maxima or have a large increase of energy associated with them. The formula given below (1) is used to determine a value for all the spectral points. If the value of the points is above a certain threshold, it is treated as a salient point.

$$Thr = \alpha * \beta = \frac{3S(x,y) - \sum_{j=-1}^{+1} S(x,y+j)}{\sum_{j=-1}^{+1} S(x,y+j)} * \frac{S(x,y)}{\sum_{i=-1}^{+1}\sum_{j=-1}^{+1} S(x+i,y+j)} \qquad (1)$$

Another alternate approach to the Shazam Technique [3] was proposed by J. George *et al* [6] which creates an algorithm which is more tolerant to time stretching. It works by identifying the 3 highest frequency bins are selected for each time slice in the spectral representation. These 3 frequency bins are then encoded into a 30-bit code, where each bin is represented by 10-bits. The audio file can thus be represented as a sequence of feature codes Ci, where 'i' is the sequence number. The search technique is based on extracting all the matching feature codes from the database and grouping them by the Song IDs. For each matched song in the database the feature codes are compared to obtain the largest common subsequence (LCS) between the two sequences.

The paper by J. Haitsma *et al* [7] proposed an entirely different Fingerprinting technique. In this scheme, the source audio is divided into frames and for each frame, a spectral representation is obtained, from which 33 non-overlapping, logarithmically separated, frequency bands are selected between 300Hz and 2000Hz. Each frame results in a sub-fingerprint and the individual bits of each sub fingerprint are calculated by the below formula (2). Sub-fingerprints by themselves don't contain enough information to match songs and hence are grouped together into blocks of 256, which can then be used for matching.

$$F(n,m) = \begin{cases} 1 \text{ if } E(n,m) - E(n,m+1) - \left(E(n-1,m) - E(n-1,m+1)\right) > 0 \\ 0 \text{ if } E(n,m) - E(n,m+1) - \left(E(n-1,m) - E(n-1,m+1)\right) \le 0 \end{cases} \qquad (2)$$

A search method based on Haitsma's method [7] was proposed by S. Lee et al [8] which proposed assigning weights to all candidate songs which contained matching sub-fingerprints. Weights are assigned to the corresponding fingerprint blocks based on the number of occurrences of matching sub-fingerprints and the standard deviation of their locations. These weights are used as a priority to speed up the search process. The fingerprint blocks are then sorted by weight and the similarity is computed between the blocks. If the computed value is below the threshold, then it is accepted.

2.3 Audio Steganography

The Message Encryption feature is based on the technique known as Steganography, more specifically, Audio Steganography. Steganography is one of the best techniques employed for data security. Steganography hides the information in such a way that, the very existence of that information is undetectable.

The paper by Sumeet Kaur et al [9] explains about Steganography and techniques involved in embedding the message into a cover medium. Steganography is also called 'Covered Writing' as it disguises the existence of the data into the cover object. Steganography provides utmost guarantee of security that no other tool can provide. There are many techniques to embed data in a cover medium and each technique uses its own mathematical approach. This makes it difficult to classify the techniques. The methods used in hiding the secret data into cover object are Spread Spectrum, which works by spreading the narrow band signal, Statistical, which uses 1-bit steganography, and Distortion, where data is embedded by the distortion of the cover.

Amba Mishra et al [10] published a paper which defines Audio Steganography as a technique of embedding a text, image or an audio into an audio file. Embedding an audio message into an audio file is much more complicated compared to hiding secret data into an image. Methods used for embedding process in an audio medium are LSB Coding, Parity Coding & Echo Data Hiding. LSB Coding is one of the simpler methods to embed secret data into an audio file. In this method, the least significant bits are substituted by the bits of secret data. This doesn't affect the cover medium in a visible way. Hence, it is one of the most widely used methods.

Along similar lines, the paper by Rohit Tanwar et al [11] explains the major parameters and some other methods used in Audio Steganography. The parameters that it defines are Perceptual Transparence, Robustness and Capacity. Other methods for Audio Steganography were explained in this paper are, Phase Coding, which is based on selecting the phase components within the original speech spectrum and replacing the components by the data to be hidden, Spread Spectrum, which makes use of a code which does not depend on the original signal and spreads the secret message along the frequency spectrum of the audio signal, and Tone Insertion, which takes advantage of limitation in human auditory system and uses the insertion of low power tones in the presence of high-power tones so that lower power tones cannot be heard.

Finally, the paper by Sangita Roy et al [12] explains the process of the LSB Encoding technique for embedding information into an audio file. LSB encoding is enabled by substituting the least significant bit of each sampling point with a binary message. This allows for a large amount of data to be encoded. Ideally, the data transmission rate in LSB encoding is 1 kbps per I kHz. However, in some cases, the two least significant bits of a sampling point are replaced with two of the message bits. For the extraction process, the receiver will require access to the sequence or group of sampling indices, which were used in the embedding process. The sender must decide on how to choose the subset of samples that contain the secret message, and communicate that data to the receiver. The sender starts at the beginning of the sound file and performs LSB encoding until the complete secret message has been embedded, leaving the remaining samples unchanged. The main advantage of the LSB encoding method is a very high watermark channel bit rate and a low computational complexity.

3. IMPLEMENTATION

3.1 Music Streaming

Audio Streaming refers to the technology of delivering real-time audio through a network connection. In our system, the UI presented to the user enables Audio Streaming in two ways. The user may either choose to play a song, or may choose to play an entire playlist.

When a user selects a song to be streamed, the server fetches the song ID from the request. A query is generated to fetch the song details and song path for the given song ID. The song is opened as a binary file stream. First 1024-bits of song are buffered in the start. As one batch of 1024-bits are buffered, a response is created with buffered data as the payload. This response is sent to the user as a stream of audio. The next 1024-bits are buffered, and this continues till the buffer data is empty. In the end, the file is closed to stop streaming as there is no data left to stream in the buffer array.

If the user selects a playlist, all the song IDs are added to a queue array. The initial index is hereby set to zero. A backend request is queried to the system to stream the audio file, with its ID and its index as the input. The server sends its response in the form of an audio stream. After finishing streaming a particular song in the queue, the index is incremented.

3.2 Audio Fingerprinting

An audio fingerprint is an acute description of some audio file that holds information, and is unique to the audio file it represents.

Firstly, a five second query clip is recorded from the device microphone. The system is queried to recognize the audio file, with input as the stream of bytes of the audio file, which was recorded. As the server receives the byte stream, it is converted to an array of audio bytes. The byte array is passed as input to the fingerprint generator. The working of Audio Fingerprinting in our application relies on the retrieval of spectrograms for deriving the desired fingerprints.

We apply a Fast Fourier Transform to procure the spectrogram of an audio file. A good frequency resolution (10.7 Hz) is needed by the fingerprinting algorithm to reduce the spectrum leakage. On the server-side execution, the sound file has to be down sampled. However, before down sampling, the frequencies above 5kHz need to be filtered to avoid aliasing. Hence, by executing the signal with a function, and a mathematical Fourier transform, we arrive at the required spectrograms.

Using the spectrogram obtained, we obtain the points on the spectrogram that are higher than those of their neighbors. Each peak point (called the anchor) is mapped to 15 other peak points (called the target). For each such mapping, a 32-bit fingerprint is generated. These fingerprints are generated based on the frequency of the anchor and the target peaks in the spectrogram and the difference in time between the two peaks. The fingerprints are then saved in the database with the corresponding ID of the song.

The database is then queried to fetch any fingerprint of a song that is matching with that of the recorded audio. To determine the correct match among the returned songs, a histogram is created, depicting the relative time difference between the query fingerprints and the song fingerprints. The peak in the histogram is chosen as the score of the song, and all scores are compared to obtain a peak score. The song which matches with the peak score is returned along with its details.

3.3 Audio Steganography

Before encryption begins, the audio file that is to be encrypted should be of .wav format. If not, the audio file is converted into .wav format. Using the wave module, we fetch parameters such as the number of frames, number of channels, sample width and the number of samples.

After the file is loaded, the encryption process beings. The masking bits are left shifted by 8 or 16 times respectively, depending on the sample-width of audio file. From these, we take the LSBs to mask them with the secret message. The smallest byte from each case will be chosen as a reference point for encryption. There is a character limit of 999 for the length of the message that can be entered by the user. The buffer is then incrementally updated with message data. Buffer length is dependent on the number of bits that are added by the message data. The buffer is right shifted to procure the LSBs. Here, the message buffer is appended to the unpacked data of the audio file. This altered sample is repacked by appending the current altered sample with the masking bits to make it into a audio file. This file, referred to as the *stego file*, is created temporarily and would be deleted after decryption. After encryption is completed, the path of the file is returned.

Decryption starts with fetching the path of the *stego file* that is created after encryption. If the song exists, the parameters of the song are fetched. The *stego file* is unpacked into a byte array. The number of bytes to recover, i.e., extract the message data is fetched by splicing the song path. The number of LSBs are extracted through masking bits by left shifting them. The buffer is incrementally updated by performing logical AND operation between the unpacked data of *stego file* and the LSBs. Here, the message data is extracted by performing a modulo operation, and is extracted in the form of ASCII. After decryption, the *stego file* is deleted, and the data string is returned.

4. ADDITIONAL FEATURES

4.1 Recommendations

The music recommendation system is a key feature of music service which generates recommendations based on the user's listening habits. Recommendation systems can be broadly classified into two types- Collaborative Filtering and Content-Based Filtering. In our system, we have implemented Collaborative Filtering.

This technique involves collecting a vast amount of information about the all the users on the system, and predicting what users would like based on their similarities to other users. A crucial difference between this technique and the Content Based Filtering technique is that the suggested music itself is not evaluated, but instead the recommendations are made based on the listening habits of the user. For example, if two users listen to the same song, the algorithm determines that the users may have a similar preference, and therefore recommends music that the other user has listened to.

Such systems require a large amount of historic data to accurately recommend music. This data may be collected in the form of user ratings or can also be derived by analyzing a user's playback history. The approach followed in the proposed system is to collect data implicitly by tracking users' listening history. Song IDs of the past 50 playbacks are stored in the database. The database is continuously updated with the song IDs of new songs. A Co-occurrence matrix is generated which, using the playback counts, assigns weights to each available song. These weights are used to select the final recommendations.

For new users, as there is no playback history present in database, recommendations would be based on popular songs. The popularity of songs is decided by the number of playbacks. This is made possible by introducing a column that holds the total number of playbacks.

4.2 Messaging System

A unique Messaging System has been developed in order to facilitate our Steganography feature. Our messaging system also helps in sharing of songs and messages among friends.

When a conversation is created, a notification channel is created and assigned to the unique conversation between the two users. The details of the channel are stored in the database for further use. The user subscribes to this messaging channel via Pusher. Pusher is an API which is used to send and receive push notifications. The messaging system is developed in a way that all messages are automatically deleted after they've been read by the receiver. After a user subscribes to the messaging channel, a request is sent to the server to fetch all unread messages. These unread messages are displayed on the chat box and once the conversation is closed, a request is sent to the server to delete all the read messages.

When a user sends a message, a request containing the message data and receiver details is sent to the server. The server then adds this message to the database. A Pusher event is sent on the sender's and receiver's personal channel and on the shared messaging channel. The pusher event sent on the messaging channel contains all the required message data, which is fetched and displayed on the chat box.

5. CONCLUSION

Music is an integral part of our lives and is much more than simply being a source of entertainment. Digital music has exploded in popularity in the past few years and this trend of growth is set to continue as computers advance. The final outcome of this project is a system that serves as an integration of multiple other systems such as Streaming, Fingerprinting and Steganography. To facilitate the better functioning of these technologies, our application has added features like Recommendations, Playlists and an independent Messaging system. By creating a seamless integration of all the various technologies, the users can try new features that they weren't previously aware of, and enjoy using them, all without having to switch between applications.

Although this system introduces a unique perspective towards Audio Streaming, Fingerprinting and Steganography, it still has its limitations and faces certain issues. Future work can focus on improving 3 key aspects- Adding the ability to dynamically adjust the bitrate of music being streamed (Adaptive Bitrate Streaming), Improving the speed of fingerprinting and song matching, and improving the speed of Steganographic encryption.

By introducing Adaptive Bitrate Streaming, any client who is experiencing slow network connection can simply start listening to a lower quality audio track instead of waiting for a higher quality audio track. The speed of the music identification process can be significantly sped up if multiple smaller databases are used instead of a single database, and if the fingerprint creation process for query songs is offloaded to the client device.

REFERENCES

[1] Kreitz, Gunnar, and Fredrik Niemela. "Spotify--large scale, low latency, P2P music-on-demand streaming." In 2010 IEEE Tenth International Conference on Peer-to-Peer Computing (P2P), pp. 1- 10. IEEE, 2010.

[2] Wu, Dapeng, Yiwei Thomas Hou, Wenwu Zhu, Ya-Qin Zhang, and Jon M. Peha. "Streaming video over the Internet: approaches and directions." IEEE Transactions on circuits and systems for video technology 11, no. 3 (2001): 282-300

[3] Wang, Avery. "An Industrial Strength Audio Search Algorithm." In Ismir, vol. 2003, pp. 7-13. 2003.

[4] Sun, Xiaoxue, Wenhui Zhang, and Deqin Chen. "Movie Retrieval Based on Shazam Algorithm." In 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), pp. 1129-1133. IEEE, 2018

[5] Shi, Jianhua, Xiaoqing Yu, Huanhuan Liu, and Wei Xiong. "Audio fingerprinting based on salient points for audio retrieval." (2013): 319-322

[6] George, Jacob, and Ashok Jhunjhunwala. "Scalable and robust audio fingerprinting method tolerable to time-stretching." In 2015 IEEE International Conference on Digital Signal Processing (DSP), pp. 436-440. IEEE, 2015.

[7] Haitsma, Jaap, and Ton Kalker. "A highly robust audio fingerprinting system." In Ismir, vol. 2002, pp. 107-115. 2002.

[8] Lee, Sunhyung, Dongsuk Yook, and Sukmoon Chang. "An efficient audio fingerprint search algorithm for music retrieval." IEEE Transactions on Consumer Electronics 59, no. 3 (2013): 652-656.

[9] Kaur, Sumeet, Savina Bansal, and Rakesh K. Bansal. "Steganography and classification of image steganography techniques." In 2014 International Conference on Computing for Sustainable Global Development (INDIACom), pp. 870-875. IEEE, 2014.

[10] Mishra, Amba, Prashant Johri, and Anuranjan Mishra. "Audio steganography using ASCII code and GA." In 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions)(ICTUS), pp. 646-651. IEEE, 2017.

[11] Tanwar, Rohit, and Monika Bisla. "Audio steganography." In 2014 International Conference on Reliability Optimization and Information Technology (ICROIT), pp. 322-325. IEEE, 2014.

[12] Roy, Sangita, Jyotirmayee Parida, Avinash Kumar Singh, and Ashok Singh Sairam. "Audio steganography using LSB encoding technique with increased capacity and bit error rate optimization." In Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology, pp. 372-376. ACM, 2012.

[13] I. Bilal, R. Kumar, M. S. Roj and P. K. Mishra, "Recent advancement in audio steganography," 2014 International Conference on Parallel, Distributed and Grid Computing, Solan, 2014, pp. 402-405, doi: 10.1109/PDGC.2014.7030779