VOL. 10, NO. 4, MARCH 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences


www.arpnjournals.com

# A SECURE CHANNEL PROTOCOL FOR MELP SPEECH TRANSMISSION USING DES-GA BASED APPROACH

Srinivasan Nagaraj[1] and G. S. V. P. Raju[2]
[1]Deptartment of CSE, GMRIT, GMR Nagar, RAJAM, AP, India
[2]Deptartment, Of CS&ST, Andhra University, Vishakapatnam, India
E-Mail: sri.mtech04@gmail.com

**ABSTRACT**

Integrity, non repudiation, confidentiality, and authentication are important entities in information security. Cryptography is the field of writing a secret code and a genetic algorithm is nothing but a revolutionary approach that is used in symmetric key and support for encryption and decryption and also they are faster and are suitable for performing huge flow of data. Many problems can be solved using genetic algorithms through modeling a simplified version of genetic processes. In this paper we developed a security algorithm using the feature of genetic algorithm to provide the security for MELP-compressed speech transmission in a noisy communication channel in conjunction with a Forward error correction code (FEC) [2]. FEC is a system of error control for data transmission, the sender adds redundant data to its messages and this method avoids retransmission of data and saves band width. We also devised a method to reduce noise during communication.

**Keywords:** cross over, XOR, MELP, FEC, noise reduction**.**

## 1. INTRODUCTION

**Single-channel Noise Reduction**

Spectral subtraction, the acoustic noise is added and so that the original signal is subtracted from the disturbed signal and it also depends on the accurate estimations of the noise spectrum.

In a single-channel noise reduction we can obtain an accurate time-variant model of the system. Noise reduction software performs noise range from a-posteriori SNR ratio and the a-priori SNR

Basic spectral noise subtraction algorithm

- $D(w) = P_s(w) - P_n(w)$

$$P_s'(w) = \begin{cases} D(w), & \text{if } D(w) > 0 \\ 0, & \text{otherwise} \end{cases}$$

- $P_s(w)$ is the noise corrupted input speech
- $P_n(w)$ is the smoothed estimate of noise spectrum.
- $P'_s(w)$ is the modified signal spectrum

**Assumption of unrelated signal and noise**

**White Gaussian noise**

It provides continuous and uniform frequency spectrum in a one frequency band and with equal power for each Hertz of the band. We can generate it a randomly by using the random number generator function, random

**1.1Forward error correction algorithms:**

The Turbo Codes algorithms can be used on squared and non-squared QAM constellations and include both Full-Turbo coding and Multilevel Turbo Coding

- Low-Density-Parity-Check Codes (LDPC) Forward Error Correcting (FEC) algorithms are available in several forms LIKE Full-Low-Density-Parity-Check Codes and QAM constellations.
- BCH error correction algorithms are available in several forms includes different forms for the encoder and decoder.

**1.2 Introduction to MELP**

It was the recently-selected U.S. federal standard for 2400 bps speech compression employs mixed-excitation linear predictive coding. MELP interprets short segments of speech as the output of a linear filter with an appropriate excitation signal. The work of the encoder is to design the filter and decide on the excitation signal and then characterize both as efficiently for binary data with frame that it contains , at the decoder, the encoded description is used to synthesize the filter and relate the excitation signal, thus generating the speech segment as in "[1]"and as in "[2]". But to provide more security, we have been proposed a method combined with Genetic approach to DES that provides to MELP during transmission over unsecured channels in this paper.

**Each MELP frame consists of 54 bits and represents 22.5 msec of speech:**

_ In this There are 25 bits are used to index the Filter's line spectral frequencies (LSF's) with MSVQ consisting of four stages - In the first stage of 7 bits and three more stages of 6 bits each.

_ Two gain parameters are used in each frame one 5 bits and the other 3 bits.

_ 7 bits are used to index the pitch parameter.

_ In the voiced frames, 8 bits representing Fourier coefficients of the excitation signal are included;

also included in voicing frames is 4- bits for band pass shaping and 1- bit to indicate when

a periodic pulse train is to be used for the excitation signal.

_In unvoiced frames, 13 bits are used for error control.

_ At last, The 1-bit sync bit is included with every MELP frame.

## 2. PROPOSED METHOD OF IMPLEMENTATION NOISE REDUCTION

**We made the following assumptions about our system**

The highest frequency that most humans can hear is approximately 20 kHz. Therefore, before the signal enters the A/D converter, it will be low pass-filtered to 20 kHz, which is also our sampling frequency. This will avoid aliasing during sampling.

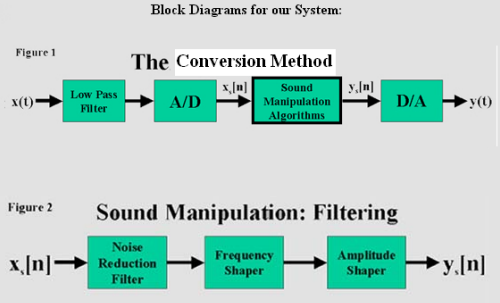


**Figure-1.** Block Diagram for NOISE reduction.

### 2.1 Implementation of Filtering

**Filter 1:** Let us, assume

1) The filter will reduce noise independent of the level of hearing loss of the user

2) That any external signals, or noise, can be modeled by white Gaussian noise.

PROCEDURE: Instead of adding white noise to a speech signal, we were able to obtain and generate several .wav sound files of a main speech signal with various sources of white noise in the background. We experimented with implementing an FIR filter, but after researching various pre-existing MATLAB commands, we used the command wdencmp, which performs noise reduction/compression using wavelets. It returns a de-noised version of the input signal using wavelet coefficients thresh holding. We also utilized the MATLAB command ddencmp.

**Filter 2: Frequency Shaper**

**Customizable design:**

Applies gain > 1 for hard-to-hear frequencies
Modifies gain for other specified ranges.
The frequency shaper is designed to correct for loss of hearing at certain frequencies. We completely designed this filter ourselves. The filter applies a gain greater than one to the frequencies that the user has difficulty hearing. As one of its parameters, the filter takes in a vector of frequencies that define the user's hearing characteristics. For each range, the frequency shaper applies a certain gain based on the user's specific hearing loss. We implemented our frequency shaper in MATLAB.

**Filter 3: Amplitude Shaper**

**The Amplitude Shaper**

We assume that the Frequency Shaper raises the frequencies that the user has difficulty hearing to sound pressure levels within his dynamic range of hearing. Therefore, all that our Amplitude Shaper has to do is check, bit by bit, that output power does not exceed a given saturation level, Psat. Since noise is concentrated in the low power levels as well, the filter also removes a significant amount of noise. Output power is equal to zero for levels below Psat.

In order to create the gain filter for sample, we used the concatenation of piecewise functions that change at f = 3000, 4000, 5000, and 9000 in hertz.

2.2. Implementation of Forward Error Correction Codes (FEC's):

There are two main properties on which a particular FEC can be selected:

• **Coding gain (Cg):** It is expressed in dB, the difference between Eb/N0 needed to achieve a given Bit Error Probability with and without encoding.

• **Coding rate (Cr):** It is the ratio of the number of message bits transmitted to the number of

**Total bits transmitted (k/n).**

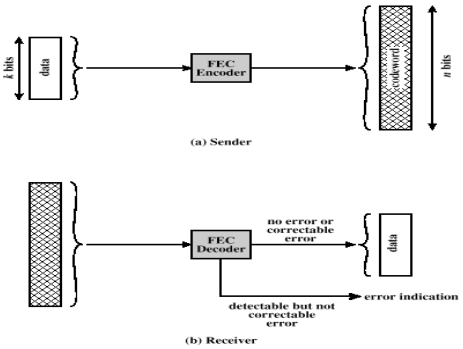


**Figure-2.** Forward error correction process.

For a receiver to correct the error with no further help from the transmitter requires a large amount of redundant information to accompany the original data
This redundant information allows the receiver to determine the error and make corrections
This error control is called forward error correction and it involves codes called Hamming codes
Hamming codes to add additional check bits to a character

– These check bits perform parity checks on various bits

• **Example: One could create a Hamming code in which 4 check bits are added to an 8-bit character**

1. We can number the check bits c8, c4, c2 and c1

2. We will number the data bits b12, b11, b10, b9, b7, b6, b5, and b3
3. Place the bits in the following order: b12, b11, b10, b9, c8, b7, b6, b5, c4, b3, c2, c1
4. c8 will perform a parity check on bits b12, b11, b10, and b9
5. c4 will perform a parity check on bits b12, b7, b6 and b5
6. c2 will perform a parity check on bits b11, b10, b7, b6 and b3
7. c1 will perform a parity check on bits b11, b9, b7, b5, and b3

• The below Figure-3 shows the check bits and their values.

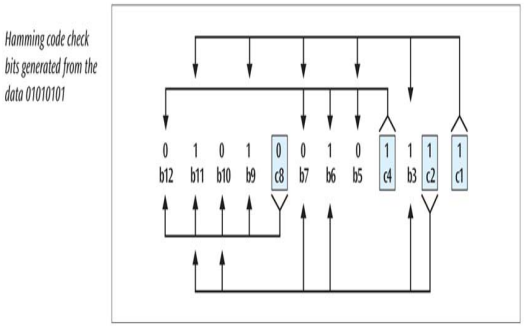


**Figure-3.** Fec correction process.

• The sender will take the 8-bit character and generate the 4 check bits as described
– The 4 check bits are then added to the 8 data bits in the sequence as shown and then transmitted
• The receiver will perform the 4 parity checks using the 4 check bits
– If no bits flipped during transmission, then there should be no parity errors
• For example, what if bit b9 flips?
– The c8 check bit checks bits b12, b11, b10, b9 and c8 (01000)
• This would cause a parity error
– The c4 check bit checks bits b12, b7, b6, b5 and c4 (00101)
• This would not cause a parity error (even number of 1s)
– The c2 check bit checks bits b11, b10, b7, b6, b3 and c2 (100111)
• This would not cause a parity error.

## 3. PROPOSED METHOD OF IMPLEMENTATION

**X9.17 generator is used to pseudorandomly generate keys and initialization vectors for use with DES**

Let $s$ be a 64-bit random seed, $m$ be an integer, $k$ be DES E-D-E encryption key, and $D$ be a 64-bit representation of time/date

1. Let $I = E_k(D)$
2. For $i = 1$ to $m$ do
    1. Let $x_i \leftarrow E_k(I \oplus s)$
    2. Let $s \leftarrow E_k(x_i \oplus s)$
3. Return($x_1, x_2, \ldots, x_m$)

Ad-hoc PRBG: FIPS 186.

### DES Algorithm

The DES (Data Encryption Standard) ASIC/FPGA core is an implementation of the DES and triple DES encryption and decryption in compliance with the NIST Data Encryption Standard [5, 7]. It processes 64-bit blocks, with one, two, or three 56-bit keys. Basic nucleus is very small (3,000 ASIC gates). A DES encryption operation transform a 64-bit block into a block of the same size. The encryption key size is 56 bits, with one to three keys used.

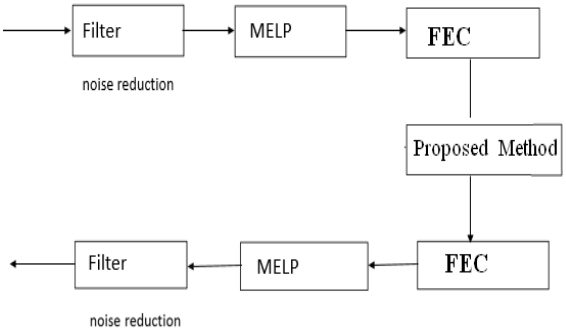


**Figure-4.** Block diagram.

**Step 1: Data Compression**

The original speech is buffered into 22.5 ms frames and passed through MELP coding filter. The 22.5 ms frame coded into 54 bits compressed speech frame.

**Step 2: Forward Error Correction**

In MELP algorithm, Forward Error Correction (FEC) is implemented in the unvoiced mode, the FEC which inputs 12 parity bits to provide error correction for voiced mode. Total 4 bits are corrected over 50 bits of data which cover all parameters except sync bit and aperiodic flag. Thus 12 parity bits are used to correct 4 errors over 50 bits out of 54 bits. This data is passed to Encryption DES –based Using GA approach.

**Step 3: Data Encryption Using Ga**

The encryption process emulates the operation of key generator and crossover operator. The encryption process comprises of following Steps:
Step I: Take 256-bits size plain text.
Step II: Apply Initial permutation to 256-bit block.

Step III: Split the plain text (256-bits) into 4 64-bits block.
Step IV: Apply cross over operation to each 64-bit block.
Step V: Using best fitted key perform X-OR operation with each 64-bit block.
Step VI: Merge the 64-bit block into 256-bit size and apply mutation operation.
Step VII: Perform 16 rounds from step3 to step6.
Step VIII: Perform inverse permutation to the 256-bits.
Step IX: Finally we will get 256-bit cipher text

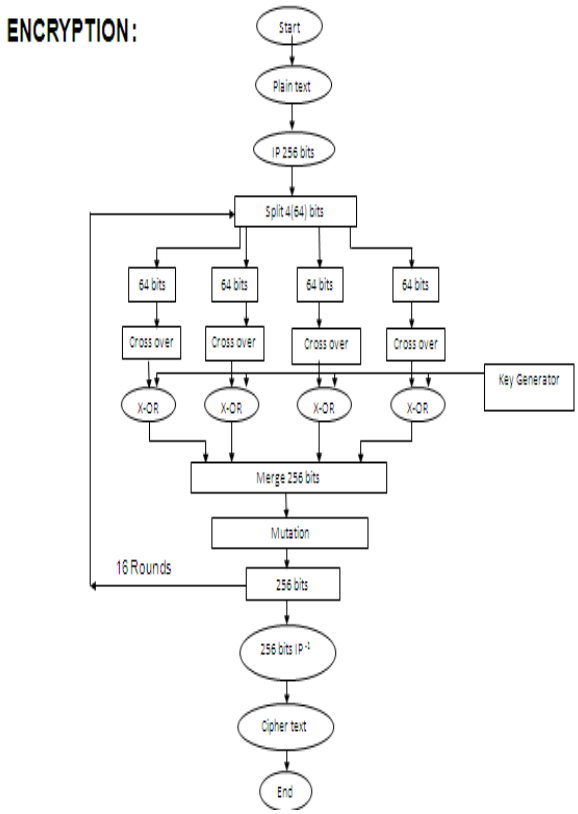


**Figure-5.** Encyption process with GA.

**Step 4: Decryption**
The steps for Decryption are just reversal of the encryption. The Decryption process consists of following steps:

Step I: Take 256-bits size Cipher text.
Step II: Apply permutation to 256-bit block.
Step III: Apply mutation operation to 256-bit block.
Step IV: Split the cipher text (256-bits) into 4 64-bits block.
STEP V: Using best fitted key perform X-OR operation with each 64-bit block.
Step VI: Apply cross over operation.
Step VII: Merge the 64-bit block into 256-bit size
Step VIII: Perform 16 rounds from step3 to step7.
Step IX: Perform permutation to the 256-bits.
Step X: Finally we will get 256-bit plain text.

**Step 5: Function of FEC**
12 parity bits are used to correct errors, if any. 54 bits of compressed speech is separated and given to MELP Decoding filter.

**Step 6: Speech Synthesis**
54 bits of compressed speech frame is passed through the MELP Decoder which produces the synthesized speech frame of 22.5 ms.

## 4. CONCLUSIONS

MELP speech coder has been selected as the U.S. federal standard for 2400 bps speech compression. The quality of MELP-compressed speech when transmitted over noisy communication channels in conjunction with a different error control schemes. In this paper we developed a security algorithm using the feature of genetic algorithm to provide the security for MELP-compressed speech transmission in noisy channels in conjunction with a forward error control, FEC avoids retransmission of data, at the cost of higher bandwidth requirements on average and we also devised a method to reduce noise during communication. Our proposed method is comparatively good performance at key generation and the confidential data is highly safe and reliable. This method provides more security because of random key generation and it is economy to develop when compared to any other public key cryptographic algorithms. To adjust for this loss, we developed a noise reduction filter in MATLAB.

In future, this work can be presently implemented with DES Approach but further extended with public key algorithms like ECC and also enhanced by making this method compatible to encrypt multimedia data which has to be transmitted securely over unsecured channels.

## REFERENCES

[1] Symmetric encryption algorithm in speech coding for defence communications. by Akella Amarendra Babu1 and Ramadevi Yellasiri ,in ITCS, SIP, JSE-2012.

[2] A New Substitution Block CipherUsing Genetic Algorithm, Srinivasan Nagaraj1, D.S.V.P. Raju2, and Kishore Bhamidipati- SPRINGER. 2013.

[3] Ye Zhu., Yuanchao Lu. and Ani Vikram. On Privacy of Encrypted Speech Communication. Dependable and Science Computing IEE Transaction. Vol. 9, No. 4, July/August 2012.

[4] Blum L., Blum M., Shub M. A simple unpredictable pseudo random number generator. SIAM J. Compute 15(2), 364–383, 1986.

[5] Cryptanalytic Attacks on Pseudorandom Number Generators John Kelsey Bruce Schneier David Wagner Chris Hal y.


[6] PENG Tan., CUI Huijuan., TANG Kun. 2010. Speech coding and transmission algorithm based on multi folded barrel shifting majority judgment; Journal of Tsinghua University. (Science and Technology).

[7] JI Zhe., LI Ye., CUI Huijuan. and TANG Kun. 2009. Leaping frame detection and processing with a 2.4 kb/s SELP vocoder, Journal of Tsinghua University (Science and Technology).

[8] Wai C. Chu. 2003. Speech Coding Algorithms, Wiley Interscience.

[9] Ghosal Prasun., Biswas Malabika., Biswas Manish. A Compact FPGA Implementation of Triple-DES Encryption System with IP Core Generation and On-Chip ,Verification. in Proceedings of International Conference on Industrial Engineering and Operations Management, 2010.

[10] Tin Lai Win. and Nant Christina Kya W. Speech Encryption and Decryption Using Linear Feedback Shift Register (LFSR). World academy of Science, Engineering and Technology. 2008.