Hartes Design Contest 2009

SIPL Team

(Signal & Image Processing Lab, Technion - IIT)

Real-Time Voice Scrambler and Descrambler

Abstract

Secure speech communication systems originated from military demands and evolved to civil and commercial uses. Speech privacy techniques are used to encrypt speech signal into an unintelligible signal in order to avoid eavesdropping. Speech data can be encrypted using analog or digital methods. Privacy communication devices comply with different security classes. The high level security class, refers to digital encryption, require large bandwidth and a digital communication channel. The lower level security class is usually limited to smaller bandwidth analog communication channels such as telephone, two-way radio, cellular devices etc [1]. Devices belonging to the later class are mainly used to encrypt less important voice transmissions or those with limited relevance time period.

The proposed design implements a complete *end-to-end* low bandwidth voice scrambling communication system. The system consists of two independent endpoints and a communication channel. Each side has scrambling and descrambling capabilities. Scrambling is performed over low bandwidth audio channels usually intended for speech. Although the incoming and outgoing signal is analog, the entire scrambling process is carried out digitally. The digital data is mathematically manipulated both in time and frequency domains.

The proposed system involves state of the art voice scrambling algorithms implemented on the Diopsis® 940HF platform. Methods that are going to be used are: time domain scrambling, frequency domain scrambling, band splitting and dummy frequencies insertion. The ARM manages encryption keys, board peripherals and manages the entire scrambling-descrambling process. All signal processing manipulations are handled by the DSP. Data interfaces such as USB or Ethernet are used to configure and control the device.

HARTES Design Description

System Description

Figure 1 depicts the layout of a scrambler/descrambler voice communication system.

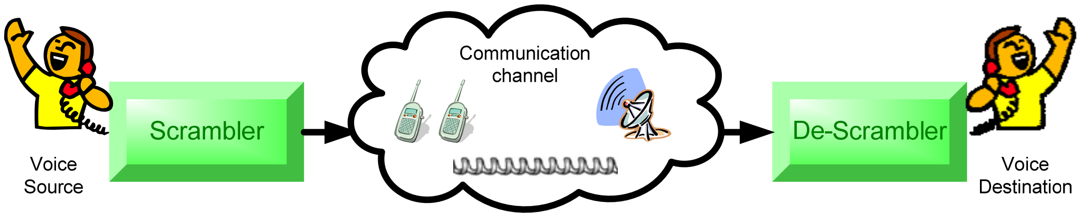

Figure 1. System layout.

A telephone-like analog audio channel connects the scrambler to the descrambler. Different analog channels can also be used with our system, as long as they have similar bandwidth. The system will be simulated running both sides on the same board, using a loopback audio connector or real telephone line. In order to simulate a real system, two development boards are required.

The Input signal is sampled and divided into frames. Each frame is going through the scrambling process independently as shown in Figure 2. The scrambling algorithm consists of several separated voice scrambling techniques, alternately in time and frequency domains. In the time domain, the frame is flipped and split. In the frequency domain, the bands are randomly flipped according to an encryption key. The main speech signal energy is usually located in low frequencies, therefore the signal's high frequencies amplitude is randomly modified in order to change the spectral envelope of the signal and strengthen the encryption. The last scrambling stage splits frames into sub-frames and transforms them individually back to the time domain. The sub-frames are being scrambled using *Bi-Dimensional keys matrix* [2]. All the generated keys in the process, are shared between the endpoints to allow recipient side descramble the signal.

Figure 3 depicts a block diagram of the descrambler. The descrambler performs similar operations to the scrambler, only in a reversed order. The voice scrambling process is not entirely recoverable [3]. In order to verify the quality of results, the descrambled signal will be compared to the input signal using standard speech quality measures such as SNR, LSD, and PESQ.

Development Phases

The development phases are:

1. Literature survey of scrambling algorithms and techniques [1].
2. Scrambling/descrambling algorithm development using MATLAB.
3. Implementation of a real-time algorithm on PC using the C programming language.
4. Code migration to the DIOPSIS 940 platform.
5. Code optimization and tweaking on the DIOPSIS 940 platform to achieve real-time performance.

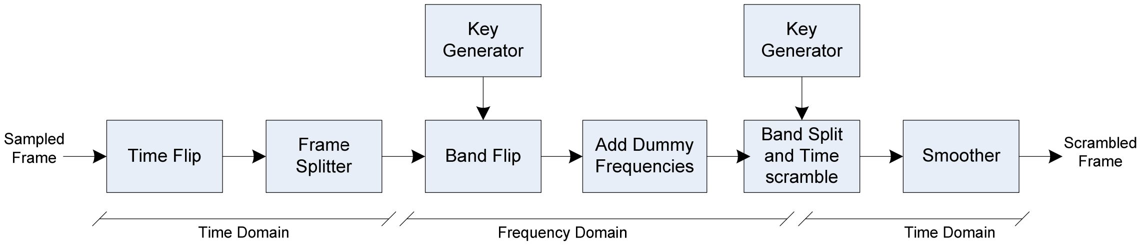

Figure 2. Scrambler Data flow.

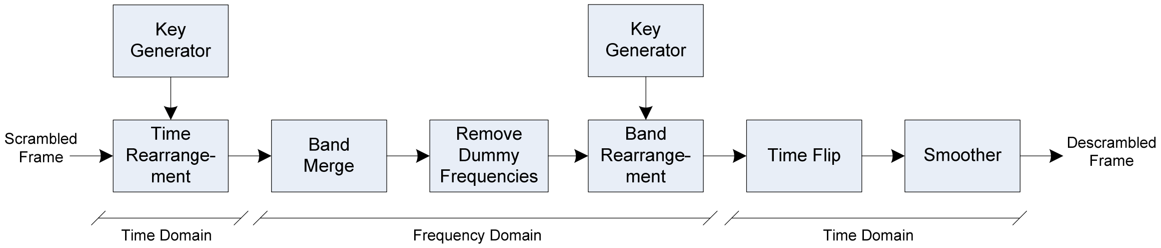

Figure 3. Descrambler Data flow.

References

1. A. Bateman, J. D. Marvill, and J. P. McGeehan, “Voice Scrambling for Radio, Cellular and Telephone Systems,” Proc. of IEEE 42nd Vehicular Technology Conf., vol. 2, pp. 968-72, 1992.

2. F. Jose, L. Marcello, and A. Jose, “Speech Privacy for Modern Mobile Communication Systems,” Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. (ICASSP’08), pp. 1777-80, 2008.

3. M. S. Ehsani and S. E. Borujeni, “Fast Fourier Transform Speech Scrambler,” Proc. 1st Int. IEEE Symp. Intelligent Systems, vol. 1, pp. 248-51, 2002.