Digital Signal Processing
Encrypted Voice Transmission with Motion Sync

Govindu Vignesh, Gosike Raviteja, Bootla Sahasrith

In our project, we created a Simulink-based Android application that can send encrypted audio and motion data in real time between two phones. Phone A, which acts as the sender, and Phone B, which acts as the receiver. The system was built entirely in Simulink Mobile, as it provides an easy way to connect different components, such as audio capture, sensors, and network communication blocks. We chose the UDP protocol for data transmission because it offers fast and efficient communication, which is important for real-time applications, even though it does not guarantee delivery of every packet.

On the sender side, the phone records audio at a sampling rate of 44.1 kHz using its two microphones. The recorded signal is processed in frames of size 4410×2 and converted into double precision for compatibility with Simulink blocks. For security, we used frequency-domain encryption through a process called frequency scrambling. The captured audio is first converted from the time domain to the frequency domain using the FFT block. Then, we multiply the frequency components by an invertible matrix P, which serves as our encryption key. This process scrambles the audio signal, making it unintelligible. At the receiver end, we use the inverse of this matrix, P^{-1} , to descramble the frequencies and reconstruct the original sound. The same key matrix is shared securely between the two devices before communication begins, ensuring that only the receiver with the correct key can decrypt the audio.

In addition to audio, the sender phone also collects motion data using the built-in gyroscope and accelerometer sensors. These sensors provide readings for the X, Y, and Z axes, representing angular velocity and linear acceleration. The six sensor values, along with a frame counter that keeps track of the data sequence, are combined into one signal. This allows the motion data to be synchronised with the audio stream. The combined data of encrypted audio, motion readings, and frame number is then sent to the receiver through the UDP Send block.

On the receiver side, the incoming data is unpacked to separate the audio, motion readings, and frame counter. The motion data is displayed in real time on the phone, while the scrambled audio undergoes decryption. It is first converted back into the frequency domain using FFT, then multiplied by the inverse key matrix to reverse the scrambling, and finally transformed back into the time domain using IFFT. The resulting real part of the signal is the recovered audio, which is then played back on the receiver phone.

Overall, our implementation demonstrates how Simulink can be used to combine real-time audio encryption, motion sensing, and wireless communication in a single mobile application. The use of frequency scrambling provides a simple yet effective form of encryption, while UDP ensures smooth real-time performance. By integrating synchronised motion data, the system can be extended for interactive or motion-based communication applications in the future.