Sonic Security: Real-Time Audio Encryption Proposal

So secure, even your FBI agent can't understand you 🕵️

Sonic Security Experts:

Jaewon Lee (jl6367), Tyler Chang (tc3407), Joshua Mathew (jm5915)

Introduction

We're keen to present a hardware-accelerated system for real-time audio encryption using the Terasic DE1-SoC development board. By leveraging the DE1-SoC's FPGA fabric, we'll create a dedicated accelerator that encrypts audio signals on-the-fly, transforming normal speech into unintelligible noise that can only be decoded with the proper key.

Whether you're discussing classified information, planning a surprise party, or just value your privacy, Sonic Security ensures your conversations stay between you and your intended audience. Just plug in, speak normally, and let our encryption make your words disappear into digital noise—no eavesdropper will make sense of it without the decryption key!!

Technical Specifications

FPGA Platform

- **Hardware**: Terasic DE1-SoC with Cyclone V FPGA and ARM Cortex-A9 HPS
- **Memory**: 85K programmable logic elements, 4,450 Kbits on-chip memory, 64MB SDRAM for buffering

Audio Processing

- **Quality**: CD-quality audio (16-bit stereo at 48 kHz)
- **Codec**: Wolfson WM8731 audio CODEC (24-bit resolution capability)
- **Data Rate**: ~1.536 Mbit/s (well within FPGA capabilities)

Encryption Engine

- **Algorithm**: AES-128 (hardware implementation on FPGA fabric)
- **Mode**: CBC or CTR for streaming operation
- **Block Size**: 128 bits (encrypting chunks of 4 stereo samples at a time)

Hardware-Software Interface

For communication between the ARM CPU and FPGA encryption accelerator, we'll implement a command-based interface where the HPS sends audio data to the FPGA through a FIFO, using DMA to minimize CPU involvement.

Audio Format

We'll use standard uncompressed PCM audio (WAV format) for file-based input/output, while live audio will come directly from the codec's ADC.

User Interface

- **KEY0**: Toggle encryption on/off
- **KEY1**: Start/stop audio processing
- **KEY2**: Select audio source (line-in vs pre-recorded test file)
- **KEY3**: Record current output to file
- **LEDs**: Display status and audio levels

Project Timeline

Week 1: Audio I/O Setup

- Configure the WM8731 codec for 48kHz, 16-bit stereo
- Implement basic audio pass-through (line-in to line-out)
- Milestone: Functional audio path with proper sample rate

Week 2: AES-128 Core Development

- Implement AES encryption algorithm in VHDL/Verilog
- Verify against known test vectors
- Milestone: AES core passing all test cases

Week 3: Integration

- Connect audio pipeline with encryption engine
- Implement buffer management and synchronization
- Milestone: Real-time encrypted audio output

Week 4: Testing and Refinement

- Test with various audio sources (speech, music, tones)
- Optimize for latency and performance
- Optional: Implement decryption mode for demonstration
- Milestone: Complete working system

Before & After Encryption Visualization

When encryption is active, clean audio waveforms transform into random noise patterns, effectively obscuring the original signal while maintaining the same amplitude range. Only with the correct key can the original audio be reconstructed.

Stretch Goals

If time permits, we'll explore:

- Multiple encryption algorithms/modes
- Variable key length support
- Remote key exchange (simulated)
- Visualization of encryption in real-time

Our project demonstrates real-world cryptography applications while providing a practical tool for secure audio communication—making privacy not just possible, but simple.