On The Voice and Image Data Encryption using Advanced Encryption Standard (AES) in Counter Mode for Multimedia Broadcasting

Junghwan Kim and Srinvasa R. Basavarasu
EECS Department, The University of Toledo, 2801 W. Bancroft St, Toledo, Ohio, 43606, U.S.A
jkim@utnet.utoledo.edu

***Abstract* - In this paper, we provide a potential solution to counter one of the shortfalls of data transmission through wireless channel, such as insecurity, by using the AES block cipher operating in stream mode, more specifically the counter mode (CTR). The approach in this work is to use the counter mode to encrypt audio and image data, to show the feasibility of implementation. The simulation results of the application of encryption and decryption confirm the effectiveness of CTR mode for successful reconstruction of audio and image only with the knowledge correct security key.**

***Index Terms*- Multimedia Quality and Content, Multimedia Processing, Data Encryption, AES Counter Mode, Security Key Algorithm**

I. INTRODUCTION

WITH the advent of broadband multimedia data transmission systems, including DAB (Digital Audio Broadcasting), WiMAX, and personal communication devices, users today are demanding a very stringent requirement on communication confidentiality and security [1, 2]. Adopting and applying various encryption algorithms can ensure security of data from spoofing and eavesdropping from the unauthorized attacks and crypto-analyst. However, the current forms of broadcasting and delivery of multimedia data through wireless channels, are highly insecure and vulnerable due to the inherent nature open access from massive users and receivers, if not properly encrypted [3, 4].

To increase the reliability and security in data communication, we provide a potential solution to counter some of these shortfalls using the Advanced Encryption Standard (AES) [5] block cipher operating in stream mode, more specifically the counter mode (CTR). AES was introduced by, National Institute of Standards and Technology (NIST) [6] in 2001. AES is adaptable in that it can be implemented on both memory-bound hardware like 8-bit microcontrollers as well as dedicated hardware to provide real-time encryption of streaming data at processing rates reaching gigabits per second.

AES supports five secure modes of operation approved by the Federal Information Processing Standard (FIPS). They are Electronic Code Book Mode (ECB), Cipher Block Chaining Mode (CBC), Cipher Feedback Mode (CFB), Output Feedback Mode (OFB), and Counter Mode (CTR) [6]. The mode of operation is crucial to the successful encryption of data for the purpose of preserving the cipher against attacks.

AES-CTR mode can be made to run exceedingly faster than other secure modes of AES without sacrificing the security requirements of a secure cipher. Additionally CTR encryption and decryption are parallelizable and simple in implementation of the counter mode. Because encryption and decryption use the same function, the code takes up less memory and reduces areas of hardware implementation. Due to these reasons, AES -CTR can be implemented in embedded applications which are in most cases, memory-bound due to the scarcity of device memory.

The remainder of this paper is organized as follows: Section II describes the encryption process. Section III contains a description of CTR mode in encryption and decryption process. Section IV contains results; and Section V concludes the paper.

II. IMPLEMENTING AUDIO ENCRYPTION

Fig.1 shows the top-level view of implementing audio encryption. The human subject's input voice into the computer. The computer's sound card encodes the analog input using Pulse Code Modulation (PCM) and compresses and stores the audio samples as a file in the form of array variable. The audio encryption is done by the AES block cipher operating in stream mode, more specifically the counter mode. We sample three outputs from this experiment.

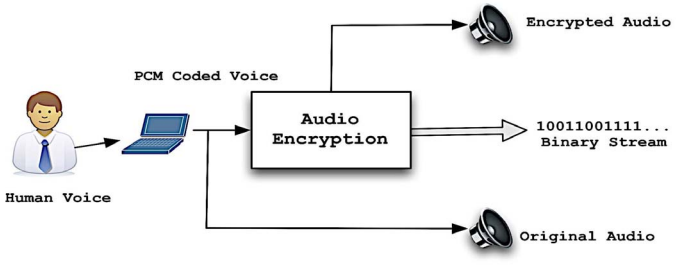

Fig. 1. Audio Encryption Expeirement

The first output is sampled from the stream of binary bits, which are packetized, after which these packets are constructed into frames. The second output is the encrypted

audio signal. This signal, when heard, sounds like a series of beeps and tones. This is due to the change in the amplitude levels and change in bit positions, affected by the cipher functions. Converting the final binary encrypted audio bits (first output) into quantized audio that is written into another files, derives the reconstructed signal. The third output is sampled directly from the input file. We compare the encrypted signal with the original audio sample in the results. We explain the implementation of the various sub-modules of the encryption and decryption implementation in the following sections.

III. AES-CTR Mode Encryption/Decryption Process of Voice and Image

A. AES Encryption/Decryption

The AES cipher has to operate in one of the defined modes according to the specifications of NIST [6] to preserve the security of the input data. We implement the Counter Mode of operation to encrypt the audio data by the use of a stream transformation function. The AES cipher operating in Counter mode serves both purposes of encryption and decryption. For this reason, we implement the same encryption engine for decrypting data at the receiver. The simplified block diagram of the counter mode implementation commonly used for the case of voice encryption can be found in Fig. 2.

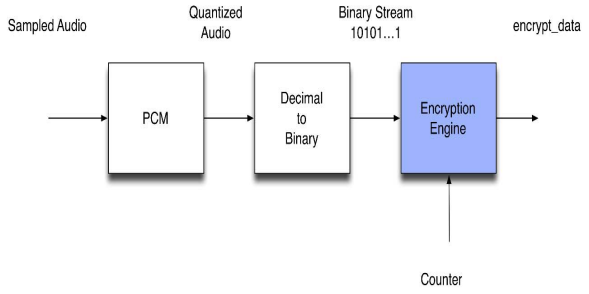

Fig .2. Top-level view of Counter Mode Implementation of Audio Signal

B. Implementing the Encryption Engine

The counter mode differs from AES block modes. Fig 3 shows the implementation of counter mode in detail. The input key(hexadecimal value) is used to encrypt the value of the counter CTR and the AES encrypted counter value is then XORed with the input binary stream. The resulting output is the binary stream. Once the key is loaded after key expansion module, then round key is also generated. The key expansion module produces 10 other keys deriving from the initial key so that the total key length is 176 bits. As shown in Fig.3, the AES process starts with the confusion and duffusion phases. It includes a series of operations of substitution, shift, and mixing. Fig. 4 shows the block diagram of the key expansion module in this regard.

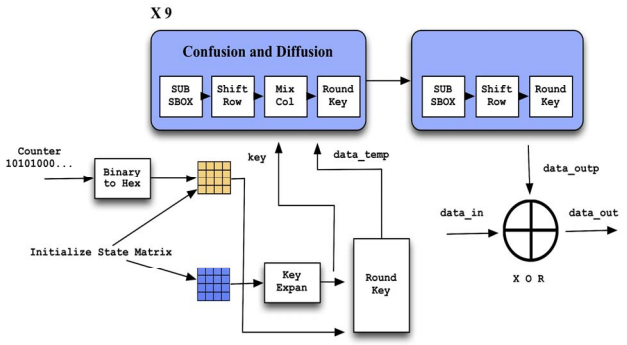

Fig. 3. Counter Mode Implementation

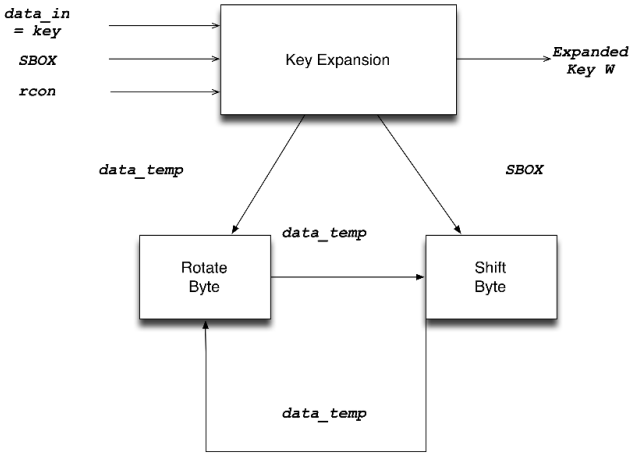

Fig. 4. Key Expansion Module pre-computed in MATLAB

For voice encryption, PCM outputs (8-bit quantizing based on 8 KHz sampling) the quantized audio. The output is fed to decimal to binary stream transform function. This function converts them into binary stream that needs to be encrypted. There are two inputs to the Encryption Engine. The first is the value held in the counter and the second is the binary stream to be encrypted. Fig. 5 shows the PCM implementation (of Fig 2) using pseudo code. After the binary to hexadecimal conversion, secret key as well as round key is attached to the data. Then they are fed to the encryption engine which is consisted of multiple confusion and diffusion operations to yield encrypted data.

```
INPUT: INPUT, Number of bits = 8, levels = 2^bits, count = 8500
OUTPUT: quantized_audio

OPEN INPUT
READ INPUT

DEFINE step size = (2*Mp)/levels
Sampling frequency 8000 samples/sec
bit rate = Sampling frequency * Number of bits
FOR K = 1 to Number of Samples (8000)
     sampled input = m(k*Ts)
     quantized input = sampled input/ step size
     error = (sampled input - quantized input)/ Number of
Samples
     k=k+1
end
quantized output = quantized input
FOR i = 1 to count
     S(i)= absolute value of (quantized input + 0.5)
     quantized output = +/- * rounded value ( S(i)* step size)
end
```

Fig. 5. Pseudo Code for PCM

For the image encryption, color image of young lion resting on a tree (Fig.10) was initially taken, read and converted to gray scale image (128x128=12384 pixel values at the individual pixel positions) and stored as an array. The gray scale integer vector of the image represents the pixel values. Then they are sent to the encryption engine of Fig. 2 after decimal to binary conversion as audio encryption. Fig. 6 illustrates the image encryption process blocks in detail.

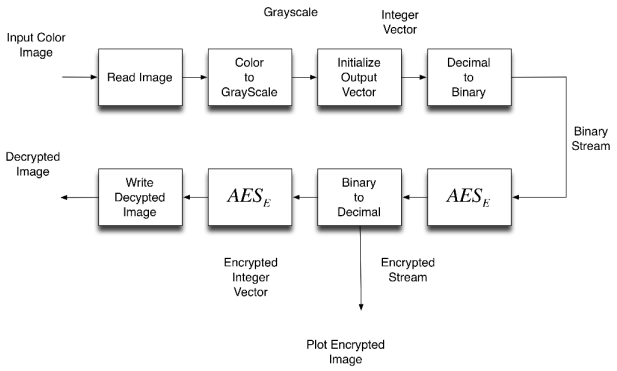

Fig. 6. Image Encryption Process

C. Decryption

The decryption process is the exact same process as encryption due to the nature of the counter mode operation. The binary encrypted stream is decrypted and converted to decimal to retrieve the original image. The decrypted binary stream is XORed with the Counter value to produce the final output. We tested this encryption/decryption algorithm with the test vectors as specified by NIST.

IV. RESULTS OF VOICE AND IMAGE ENCRYPTION

For the purpose of simulating encryption using Matlab, we took two input sources, an input audio signal and input image. The input audio signal sounds "This is a test signal" and the image is a young lion resting on a tree respectively.

Per voice, Fig. 7 shows the difference between waveforms of the audio input and the AES encrypted audio output. We extracted this plot from the output of the encryption engine. It is simple to observe that the encrypted output waveform looks very different from the original audio samples and it is impossible to deduce the original. It is also interesting to compare the number of samples as a function of amplitude in Fig.8. In the input clip we can see that there is a higher concentration of samples at lower amplitudes. This is obvious because of the many low amplitude values that occur in human speech. The highest amplitude is at 0.6. The encrypted samples shown in Fig.9 on the other hand, are distributed somewhat equally between the maximum and minimum values, according the confusion and diffusion properties of the AES algorithm.

The attacker cannot retrieve the input signal based on the information contained in the encrypted output. Fig.9 shows the bar graph of the original audio input binary values and bit positions before encryption, whereas Fig.10 shows the corresponding one after audio encryption.Fig 11 shows the comparison of these two figures. As can be seen, distributions of input and output bit streams of AES-CTR engine are totally different. The encrypted binary stream has many more '1's than the input stream. It is impossible to even pinpoint the bit values based on the time position of the samples. We also interlaid both streams and found that there is no resemblance between the input and output binary streams and the attacker cannot reconstruct the audio sample from the encrypted binary stream.**Per image,** the grays scale image that we used as input to the encryption engine is shown on the left most side of Fig. 12. The encryption engine substitutes and shifts pixel values as specified by the routines Sub Bytes and Mix_Column of the AES algorithm. The encrypted output image looks just like an image corrupted by noise (Center of Fig.12). Even with the knowledge of individual bits, it is practically impossible to reconstruct the original image from the output. With the knowledge of the correct key, the encrypted image is decrypted and the final output at the decryption engine (which is the same as the encryption engine because of CTR Mode) is obtained as shown in the right side of Fig.12. It is identical to the original image. In Fig. 13, we show that the pixel intensities along with the pixel positions are changed.

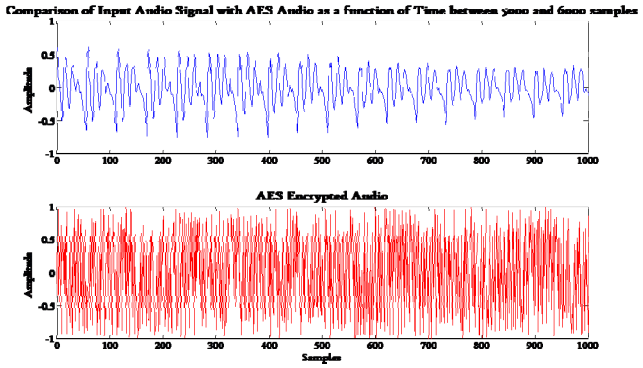

Fig. 7. Comparison of Input and AES Encrypted Audio Waveforms

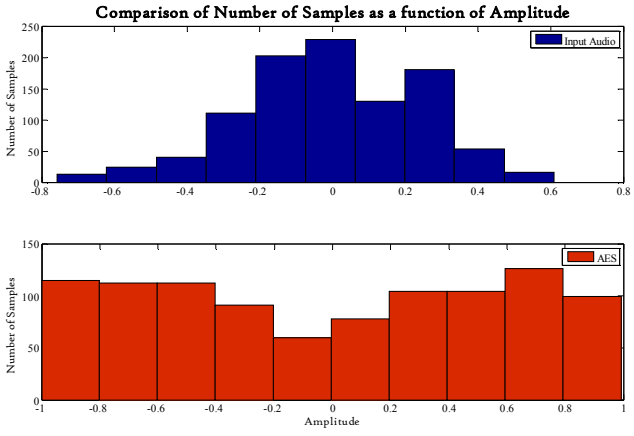

Fig. 8. Comparison of Input and Encrypted Audio of Number of Samples as a function of Amplitude

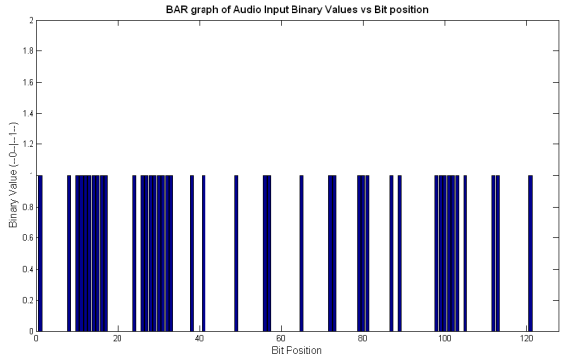

Fig. 9. Bar graph of the Original Audio Input Binary Values vs. Bit Position

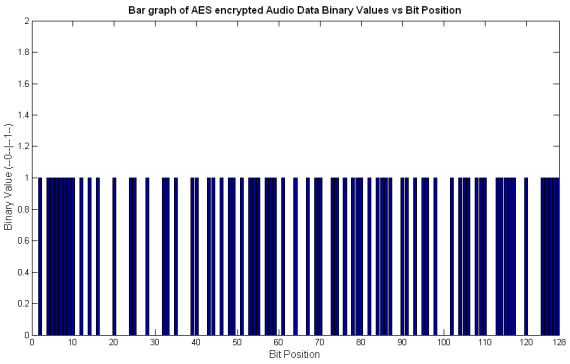

Fig. 10. Bar graph of AES encrypted Audio Binary values vs. Bit Position

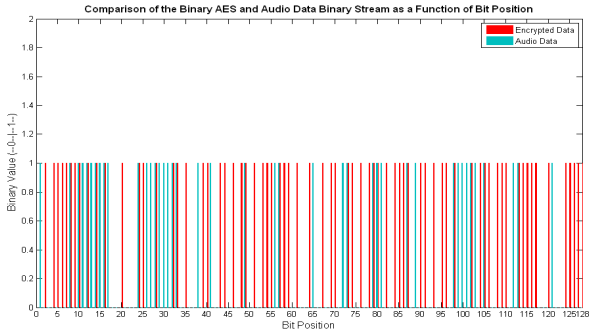

Fig. 11. Comparison of bit positions of binary audio data streams before and after encryption

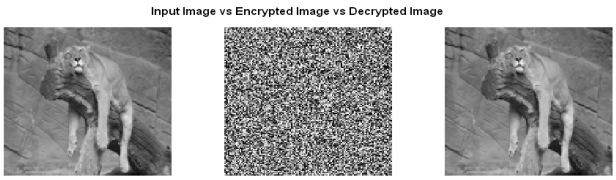

Fig. 12. Input Image (Left) vs. Encrypted Image (Center) vs. Decrypted Image (Right)

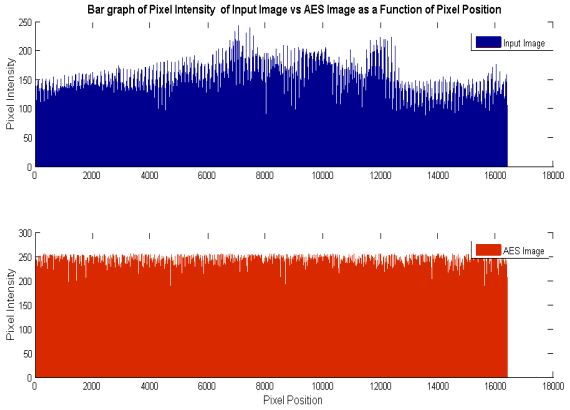

Fig. 13. Bar graph of Pixel Intensity of input image vs.AES Image as a function of Pixel position

V. Conclusion

In this paper, voice and image data have been successfully encrypted using AES - CTR mode. The difference between waveforms of the audio input and the AES encrypted audio output were compared. Results show the impossibility to deduce the original input audio. Also the encrypted image showed that even with the knowledge of individual bits, it is practically impossible to reconstruct the original image from the output.

References

[1] T. Shon and W. Choi, "An analysis of mobile WiMAX security: vulnerabilities and solutions," in Network-Based Information Systems. vol. 4658, Springer, 2007, pp. 88-97.

[2] Nguyen, T., "A survey of WiMAX security threats," Computer Science Department, Washington University, 2009.

[3] Mao, W., "Modern Cryptography: Theory and Practice," 2003: Prentice Hall Professional Technical Reference. 740.

[4] D.R. Stinson, "Cryptography: theory and practice," CRC press, 2006.

[5] J. Daemen, and V. Rijmen, "The design of Rijndael: AES-the advanced encryption standard," Springer, 2002.

[6] FIPS 197, Advanced encryption standard (AES). National Institute of Standards and Technology, 2001.