Voice Encryption and decryption using AES Algorithm

Manthan Patil[1], Ankita Mohokar[2], Supriya Gorkha[3], Rinku Yadav[4], Prof. W. P. Rahane[5]

Students, Information Technology, NBN Sinhgad School of Engineering, Ambegaon BK., Pune[1,2,3,4]

Associate Professor, IT, NBN Sinhgad School of Engineering, Ambegaon BK., Pune[5]

Abstract: *Cryptography assumes a vital job in security of information transmission. The improvement of registering innovation forces more grounded necessities on the cryptography plans. In 2000, the Advanced Encryption Standard (AES) supplanted the DES to conquer the expanding prerequisites for security. In cryptography, the AES, likewise called as Rijndael, is a square figure that is received as an encryption standard by the USA government, which determines an encryption calculation fit for securing private and touchy data. This calculation is a symmetric square figure that can encode and unscramble the data.*

Keywords: AES Algorithm, Cryptography, Encode and Unscramble the Data, etc.

I. INTRODUCTION

It is a research-based project about implementation of Voice Encryption and decryption using AES Algorithm. Encryption is an efficient method for protection of speech communications. Voice Encrypt or digitize the conversation at transmitter end and apply a cryptographic technique to the resulting bit-stream. In order to decipher the speech correct encryption scheme must be used. Voice Encryption helps us in private and confidential manner. It is nearly impossible to decrypt voice into its original form again. We will compare the efficiency and performance of standardized C/C++ implementations and open-source Verilog codes with ours and compare the results.

We will first write code for our algorithm in higher level language usually in c or C++. The C implementation will be synthesized into its equivalent Verilog implementation using higher level synthesis tool. Xilinx VIVADO HLS can convert into a higher language code into its parameterized Verilog modules. After that performance comparison will be done in terms of efficiency and latency. A public key and private key will be used. It's a choice that whether public key or private key is being used on sender end and receiver end will use the alternate key. Private Key should not be compromised.

The project will operate in Visual studio, ISE design and VIVADO HLS, including the hardware platform of Spartan 6. Project is about giving a speech signal as input to an Analog to Digital Converter (ADC) at the transmitter end which will convert voice into its digitized form and then voice samples will be stored it on FPGA Block rams. The voice will be decrypted at transmitter end. After sending it through a wired communication channel (Ethernet cable), it will be received on second FPGA the data will be decrypted and later convert from binary to speech signal through Digital to Analog Converter (DAC).

II. LITERATURE SURVEY

[1] Existing Voice encryption techniques are implemented in C/C++ and Verilog. We can use these existing techniques for our research and convert the existing C/C++ code in Verilog by synthesizing it by VIVADO then comparing it to existing Verilog and C/C++ techniques by speed testing, complexity and security. As Rijndael has existed as the Advanced Encryption Standard since 2000, there are already many implementations of the algorithm. These implementations are available in many different programming languages. As the AES standard is open, organizations or users which wish to implement the Rijndael algorithm are free to do so.

DOI: 10.48175/IJARSCT-5273

[2] Encryption techniques as far as hardware implementations are concerned can be broken down into following categories:

a) Implementation in C/C++
b) Implementation in Java
c) HDL based Implementations
d) GPU based implementation.

Unwavering quality, openness, mystery and secretly of correspondence are the primary angles that would be kept up in voice security. Shielding voice frameworks from disturbance and adjustments just as the illicit access is the primary objective of voice security. A productive and secure correspondence framework for voice flags that will base on AES open key cryptosystem is structured in this work. The introduced cryptosystem is executed and its execution is assessed utilizing diverse voice quality measurements in both encryption and unscrambling forms.

[3] As previously stated, this project requires the implementation of the Rijndael's algorithm in systems constructed with several different programming languages. Therefore, it is important to examine the history and nature of these languages, so as to understand how they might be used and how cryptography might be used in a system developed in that language. The languages to be used in this project are Java, C, JavaScript and Perl. Java -> Java is an object-oriented language that was created and developed by software engineers at Sun Microsystems during the early 1990s. Code written in Java is compiled into byte code which can then be executed on the Java Virtual Machine. In practice this means that programs written in Java can be run on any computer which supports a JVM, regardless of the underlying hardware or operating system, which is a key feature of the language. While Java can be used to create desktop applications, it can also be used to create Java applets. These are applications which can be distributed over the internet and run within a web browser. Therefore, a potential application for cryptography would be an applet which encrypts data on the client's computer before submitting it to a server.

[4] The programming language C was originally designed in 1972 and used as the systems language for UNIX. It gradually evolved in the years following until in 1990 the American National Standards Institute (ANSI) set the standard for the language, known as ANSI C. C is a relatively low-level programming language that gives the programmer more control over aspects such as memory allocation. Like Java, C has been widely used and has a variety of potential applications, but for this language, a suitable program might be one that encrypts or decrypts files from a command prompt.

[5] Perl is a scripting language like JavaScript, and grew because of its strengths in text manipulation. Perl is often regarded as a general-purpose programming language which can be used to automate a variety of different tasks. While there is no 'typical' Perl system, one which used cryptography might do so for the encryption of data being passed between other systems. Perl is a general-purpose programming language originally developed for text manipulation and now used for a wide range of tasks including system administration, web development, network programming, GUI development, and more. Perl is an interpreted language, which means that your code can be run as is, without a compilation stage that creates a non-portable executable program. Traditional compilers convert programs into machine language. When you run a Perl program, it's first compiled into a byte code, which is then converted (as the program runs) into machine instructions. So, it is not quite the same as shells, or TCL, which are strictly interpreted without an intermediate representation. It is also not like most versions of C or C++, which are compiled directly into a machine dependent format. It is somewhere in between, along with Python and awk and Emacs etc. files.

[6] JavaScript is a scripting language that was created in the early 1990s. It is commonly used on the internet to provide client-side functionality in web pages. It has no technical relation to Java, although both languages have a similar syntax.

DOI: 10.48175/IJARSCT-5273

A system using JavaScript on the internet might make use of cryptography when displaying encrypted content to a user. JavaScript is a scripting or programming language that allows you to implement complex features on web pages every time a web page does more than just sit there and display static information for you to look at displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc. you can bet that JavaScript is probably involved. It is the third layer of the layer cake of standard web technologies, two of which (HTML and CSS) we have covered in much more detail in other parts of the Learning Area. The core client-side JavaScript language consists of some common programming features that allow you to do things like:

1. Store useful values inside variables. In the above example for instance, we ask for a new name to be entered then store that name in a variable called name.
2. Operations on pieces of text (known as "strings" in programming). In the above example we take the string "Player 1: " and join it to the name variable to create the complete text label, e.g., "Player 1: Chris".
3. Running code in response to certain events occurring on a web page. We used a click event in our example above to detect when the label is clicked and then run the code that updates the text label.

[7] The implementations are done in C, C++, Perl, Java and JavaScript is purely for simulation purposes and have no relevance to hardware in general. The category C and Java involve no usage of hardware but in GPU based implementation are still in beginning stage. So, we are left with one option that is HDL based implementation. Current project will minimize the gap of C and HDL as main algorithm blocks will be done in higher level language will be ported to synthesize HDL so that they can function as per requirement. The implementation done in higher level language is just for simulation or observation and has no relevancy to hardware design constraints in general. So, we will first take c blocks of the proposed encryption technology. VIVADO has the capability to convert the c language code into a synthesizable Verilog code.

[8] Encryption has a strong presence in today's digital electronics with the frequent transmission and storage of sensitive data. AES as the standard encryption algorithm and it is commonly used as a fast solution to secure data. When designing VLSI systems, the task of balancing the area, power, and speed is a challenge; hardware encryption is no different. System requirements drive certain performance parameters to the forefront, identifying how to alter design implementations to meet performance requirements is not always apparent. Multiple resources in this research field have identified AES algorithm features of interest and discussed their impact a few of the design trade spaces, however, a single comparative analysis was lacking. This project explores six different AES features key size, mode specificity, round key storage, round unravelling, SBOX implementation, and pipelining. A summarized view of the resulting designs allows readers to quickly analyse how each of the six features impacts speed, power, area, latency and throughput.

III. PROPOSED SYSTEM

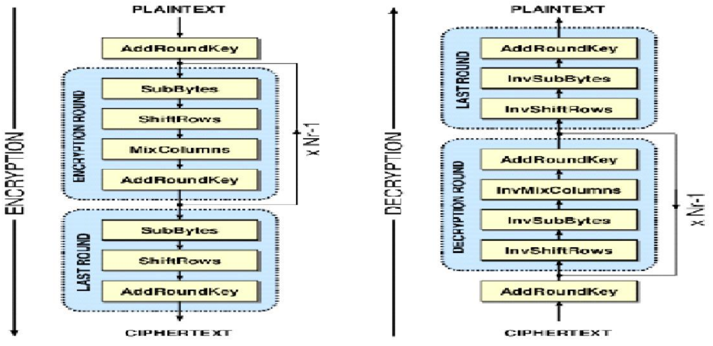

Figure 3.1: Proposed system block diagram

Introduction

Our proposed project is about implementation of Voice Encryption and Decryption over IP Networks using AES Algorithm. Encryption is an efficient method for protection of speech communications. Voice Encrypt and digitize the conversation at transmitter end and apply a cryptographic technique to the resulting bit-stream. For decrypting the speech correct encryption scheme must be used. Voice Encryption helps us in private and confidential manner. It is nearly impossible to decrypt voice into its original form again

Block Diagram

The ECB (Electronic Code Book) mode of operation is the simplest of all. A block scheme of this mode is presented in Figure. As it can be seen from Fig. the plain text message is divided in blocks (P1, P2, PN), where each block is encrypted separately with the same key (K). The results of the encryption are the encrypted messages C1, C2 and CN respectively. If the size of the message is larger than n blocks, the last block is filled with padding. In this mode, if an error occurs in one of the blocks, it is not propagated to the other blocks, which is why decryption is possible in the blocks that don’t contain an error. The encryption in this mode is deterministic, because identical P blocks will produce identical C blocks, which is why identical plain text blocks or a message with the same beginning are easily recognizable. Also, the ordering of the C blocks can be changed without the receiver noticing. In general, this mode is not recommended for encryption of data that is larger than one block.

Design Methodology:

Following Steps are followed for design methodology.
Step 1 - Taking a C language code of AES encryption and decryption.
Step 2 - Simulation
Step 3 - Converting C language code to Verilog code.
Step 4 - Comparing existing codes to new generating code.
Step 5 - Performance and execution time checking.

AES Modes of Operation

A mode of operation describes how to repeatedly apply a cipher's single-block operation to securely transform amounts of data larger than a block. Five modes of operation of the AES algorithm were standardized: ECB (Electronic Code Book), CBC (Cipher Block Chaining), CFB (Cipher Feed Back), OFB (Output Feed Back) and CTR (Counter).

CBC (Cipher Block Chaining)

In order to provide cryptographic security, every encryption of the same plain text should result with a different cipher text. The CBC (Cipher Block Chaining) mode of operation provides this by using an initialization vector labelled as IV. The IV has the same size as the block that is encrypted. Figure presents the encryption process. First, an XOR operation is applied to the plain text block (P1) with the IV, and then an encryption with the key (K) is performed. Then, the results of the encryption performed on each block (C1, C2, … , CN-1) is used in an XOR operation of the next plain text block PN which results in CN. In this way, when identical plain text blocks are encrypted, a different result is obtained. Also, using a different IV for each new encryption, an identical message will always be encrypted differently.
It should be emphasized that the same key K is used in each of the encryption blocks an error in one of the plain text blocks will propagate in all the following blocks and will be manifested in the process of the description. Specifications is recommended that the Padding method 2 is used in case padding is needed with the CBC mode of operation because it provides protection from some of the known PA (Padding Attacks). There are complex CBC attacks for which an unpredictable value of IV is needed in order to overcome them. In it is emphasized that the CBC mode of operation is safe from CPA (Chosen Plaintext Attack) attacks (attacks in which the attacker chooses a set of plaintexts and is able to obtain respective cipher texts) only if the IV has a random

value, but not if the IV is a nonce (a number that is not repeated). The CBC mode of operation, besides its vulnerability to PA attacks, is also easily susceptible to CCA (Chosen Cipher text Attack) attacks (where the attacker chooses a set of cipher texts and is able to obtain respective plain texts).

OFB (Output Feed Back)

The OFB (Output Feedback) mode of operation (Fig. 3.9) also enables a block encryptor to be used as a stream encryptor. As shown in Figure 3.9, the difference between the CFB and OFB mode is such that, in the case of an OFB, as an input for the shift register from the next block, the output from the encryptor (Encrypt) from the previous block is chosen. At the same time, the XOR operation with the s-bits of plain text P uses only s bits from the encryptor. Encryption and decryption are the same operation. If there is an error in a block during the encryption, while performing the decryption, it will influence only a part of the plain text that will result from that block, i.e., there is a limited propagation of error. Therefore, this mode of operation is often used in communication through media that carry noise (for example, satellite communications).

CTR (Counter)

At the CTR (Counter) mode of operation, shown on Fig. 3.10, as an input block to the encryptor (Encrypt), i.e., as an IV, the value of a counter (Counter, Counter + 1, …, Counter + N 1) is used. The counter has the same size as the used block. The XOR operation with the block of plain text (P1, P2, …, PN) is performed on the output block from the encryptor. All encryption blocks use the same encryption key K. If the last block of clear text PN has the number of bits smaller than the number of bits in the block, then only the s most significant bits for the XOR operation on the block PN are used from the output block of the encryptor.

The remaining bits are discarded. Hence, as it is pointed out in [4], there is no need for adding bits (padding) to the last block. Values of the counters are independent from the output of the previous block; therefore, there is no propagation of error from one block to another. Considering the independence of the blocks, this fact allows for parallelism in the encryption and the decryption, and there is also the possibility of pre- processing the values of the encryptors, which speeds up the process.

IV. IMPLEMENTATION

Development Methodologies

Existing Voice AES encryption techniques are implemented in C/C++ and Verilog. We can use these existing techniques for our research and to convert the existing techniques C/C++ code in Verilog then verification and comparison to existing Verilog and C/C++ techniques by speed testing, complexity and security.

Modules in C++ Code

AES_ctr128_axis: The module for the complete AES-128 CTR mode encryptor / decryptor, including key expansion. In this module has a 128-bit input data, 128-bit key and 256-bit of initialization vector. This module will XOR ECB (Electronic Code Book) output with plaintext for encryption and with cipher text for decryption.

AES_ecb128:

It is a complete AES-128 ECB mode encryptor, including key expansion. There are 10 rounds required for AES-128. It performs process of sub-Bytes, shift Rows, mix Columns and add round key.

ENC-MixCol:

Every input and output of different stages in this module are of 8 bits. In round 1 the four numbers of one column are modulo multiplied in Rijndael's Galois filed by a given matrix. The MixColumns step along with the ShiftRows steps is the primary source of diffusion in Rijndael.

KeyExpEngine:

Key expansion range in this module is 32 bits. This module calculates the next round key for AES key expansion.

DOI: 10.48175/IJARSCT-5273

Non-Count:
Nonce and counter for AES counter mode. Initialization Vector will be loaded at power-on/reset or at first 16byte word (message block) of new plaintext/cipher text message. The last word of current plain text/cipher text message is indicated by "last". At the first word of a new message, the counter and nonce will load from init_vect. This allows the counter to be started at a non-zero value chosen by the user. The sizes (in bits) of the counter and nonce variables can be modified but the sum of their lengths must be 128.
rotWord: Word rotate function for AES key expansion.

RCON:
RCON lookup table for AES key expansion. Expansion of the given Cipher key into 11 partial keys, used in initial round, the 9 main rounds and the final round. The expanded key can be seen as an array of 32-bit columns numbers from 0 to 43. The first four columns are filled with the given Cipher key.

Shift-Rows:
Input and output range in this module is 8 bits. It rotates over 1st byte, 2nd byte and 3rd byte.

Sub-Bytes:
In the Sub Bytes step, each byte in the matrix is updated using an 8-bit substitution box, the Rijndael S-box. This module provides the non-linearity in the cipher. The S-box used is derived from the multiplicative inverse over GF (28), known to have good non-linearity properties. To avoid attacks based on simple algebraic properties, the S-box is constructed by combining the inverse function with an invertible affine transformation. The S-box is also chosen to avoid any fixed points, and also any opposite fixed points.

SubWord:
SBOX function for key expansion. Implementation Tools and Technologies Microsoft Visual Studio was used to analyze the existing techniques; The C implementation will be synthesized into its equivalent Verilog implementation using higher level synthesis tool. VIVADO HLS can convert into a higher language code into its parameterized Verilog modules. A Xilinx ISE test tool was used for testing, verification and comparison of new generated technique with existing techniques.

Figure: VIVADO HLS

V. SUMMARY

In this project we have implemented AES technique in C/C++ and C implementation will be synthesized into its equivalent Verilog implementation using higher level synthesis tool. We use existing techniques for verification and comparison to implemented Verilog and C/C++ techniques by speed testing, complexity and security.

VI. RESULT AND DISCUSSION

We have proposed a biometric authentication and authorization gadget for growing the tool protection and decreasing the danger of cyber-assaults. Biometric identity permits quit-customers to apply as a steady technique of gaining access to a gadget or a database [9]. Biometric generation is primarily based totally at the idea of replacing one component you've got and who you are, which has been visible as a more secure generation to maintain non-public statistics. The opportunities of making use of biometric identity are genuinely substantial. Biometric identity is carried out these days in sectors in which protection is a pinnacle priority, like airports, and can be used as a way to manipulate border-crossing at sea, land, and air frontier.

Especially for the air visitor’s region, in BOB "Hello Alice!" Bob drafts a note to Alice. He gets Alice’s Public key SERVERAN83D&10B? KA%)2D The message passes through an intermediary ALICE "Hello Alice!" Alice receives the message and uses her 5 which the wide variety of flights could be extended via way of means of 44%, the authentication of cell IoT gadgets could be done whilst the bio functions fashions emerge as sufficiently mature, green, and proof against IoT assaults. Another region in which biometric identity techniques are beginning to be followed is digital IDs.[10] Biometric identity playing cards inclusive of the Estonian and Belgian country wide ID playing cards had been used in an effort to discover and authenticate eligible citizens at some point of elections. Moving one step in addition, Estonia has added the Mobile-ID gadget that lets in residents to behaviour Internet vote casting and combines biometric identity and cell gadgets. This gadget that turned into pretty revolutionary whilst it turned into to begin with, possesses numerous threats to the electoral process and has been criticized for being insecure According to a survey via way of means of Javelin Strategy & Research, in 2014, sixteen billion turned into stolen via way of means of 12.7 million folks that had been sufferers of identification robbery withinside the US best.

This quantity is calculated without contemplating the monetary troubles and mental oppression that sufferers of this fraud suffer. From the banking zone and groups to get entry to homes, cars, non-public computers, and cell gadgets, biometric generation gives the best stage of protection in phrases of privateness and privateness safety and steady entry to. Mobile gadgets are a vital part of our regular life, as they're used for a whole lot of cell programs. Performing biometric authentication through cell gadgets can offer a more potent mechanism for identification verification as the two authentication elements, that are “something you've got" and “something you are," are combined. Several answers that encompass multibiometric and behavioural authentication systems for telecom carriers, banks, and different industries have been added these days. End-to-end encryption (E2EE) is the first-rate-regarded manner to shield customers virtual communications, because it prevents provider vendors in addition to unassociated 0.33 events from analysing messages in recent years, numerous famous messaging apps have followed end-to-end encryption, both via means of default (WhatsApp, iMessage) or as an optionally available feature (Facebook Messenger, Telegram).

As a result, after many years of use best in area of interest programs and communities, E2EE is now without difficulty to be had and utilised by tens of thousands and thousands or maybe billions of customers. Many authentication schemes primarily based totally on bio functions fashions for cell IoT gadgets were proposed. The schemes can carry out distinctive authentication operations: they both are (a) authenticate the customers to get entry to the cell gadgets or (bauthenticate the customers to get entry to faraway servers thru cell gadgets. The primary demanding situations which are dealing with biometric-primarily based totally authentication schemes are a way to layout an authentication mechanism this is unfastened from vulnerabilities, which may be exploited via way of means of adversaries to make unlawful accesses, and (2) a way to make sure that the person’s biometric reference templates aren't compromised via way of means of a hacker on the tool stage or the faraway-server stage.

DOI: 10.48175/IJARSCT-5273

VII. CONCLUSION

An IoT tool has the ability for acting many obligations in a redundant and sturdy way in which people can't enter, for example, excessive temperature and faraway region manipulation/ surveillance in lots of industries rescue missions. An IoT controller has Internet connectivity that permits it to transmit faraway records to the cloud and assist examine it. The fundamental precept in the back of the controller is, the sensor and the actuator constantly talk with the controller and the controller to the cloud through MQTT Protocol. While taking the records with the assist a sensor we are able to stay streaming and seize datasets which are critical. We have proposed a biometric authentication and authorization gadget for growing the tool protection and decreasing the danger of cyber-assaults. The authentication is detected nearly immediately (within 0.5 sec). However, it takes a lot longer for the danger to get in effect, so the tool nevertheless might not be capable of running into problems.

REFERENCES

[1] Z. Bouida et al., "Carleton-Cisco IoTTestbed: Architecture, Features, and Applications," 2021 IEEE Globec Workshops (GC Wkshps), 2021, pp. 1-6

[2] C. Tharun, C. Rithin and B. Bharathi,"Double Door Authentication for Mobile Devices using Personalised Lock (pins), 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), 2020, pp. 296-301

[3] W. Kezhong, W. Yutao, Z. Ruicong, Y. Rundong and B. Yu, "A Lightweight authentication method between homogeneous nodes in Wireless Sensor Network based on Message Authentication Code," 2020 IEEE 8th International Conference on Smart City and Informatization (iSCI), 2020, pp. 68-72.

[4] T. M. Bandara, W. Mudiyanselage and M. Raza, "Smart farm and monitoring system for measuring the Environmental condition using wireless sensor network - IOT Technology in farming," 2020 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA), 2020, pp.1-7

[5] A. -E. Bouaouad, A. Cherradi, S. Assoul and N. Souissi, "The key layers of IoT architecture," 2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech), 2020, pp. 1-4

[6] J. Pei, J. Dang and Y. Wang, "Encryption method of privacy information in student archives based on blockchain Technology," 2021 6th International Conference on Smart Grid and Electrical Automation (ICSGA), 2021, pp. 208-211

[7] S. Chiţu, D. C. Vasile, I. Daniel Trămîndan and P. Svasta, "Key Expansion in Cryptographic Systems," 2020 IEEE 26th International Symposium for Design and Technology in Electronic Packaging (SIITME),2020, pp.202-205

[8] B. Chen, L. Wu, N. Kumar, K. -K. R. Choo and D. He, "Lightweight Searchable Public-Key Encryption with Forward Privacy over IIoT Outsourced Data," in IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 4, pp. 1753-1764, 1 Oct.-Dec.2021.

DOI: 10.48175/IJARSCT-5273