

OPTIMIZATION OF AES ALGORITHM USING HARDWARE AND SOFTWARE

Pavithra.S, Vaishnavi.M , Vinothini.M, Umadevi.V

Abstract-In today's world, security is very fundamental and significant issues of data transmission. Technology advancement is occurring daily in order to find a new cryptographic algorithm. Data security is concerned with the areas of data transmission. Recent advancements in cryptography has led to new techniques called Gray code based cryptography and Quantum Cryptography. In existed AES method is well suited for security applications. In our proposed system we improve the performance of the system we propose Triple AES based gray key cryptography. The XMEGA AES Crypto Module implements the Advanced Encryption Standard (AES), and can perform encryption and decryption. In our new proposed implementation provides better and faster results in environment, hence data security is high.

Index terms – AES algorithm, Encryption, Decryption, Cryptography, Gray code, Security, Key generation

1 INTRODUCTION

Nowadays technology is developing at a rate faster than speed of light and the amount of redundant digital multimedia signals has increased more in the internet. The security of a system is essential nowadays. The number of threats a user is supposed to deal grows exponentially, with the growth of Information Technology power.

Cryptographic systems have been in use for many years from now on. That is cryptography consists in processing plain information by applying a cipher and obtaining encoded output, which would seem meaningless to a third party who does not know anything about the key involves one or more keys. In secret key cryptography, secure key exchange through public channel is difficult. These shortcomings in traditional cryptography are overcome by Quantum Cryptography (QC).

The word cryptography comes from two Greek words meaning "secret writing" and is the art and science of concealing meaning. The basic component of Cryptography is cryptosystem. Basically transmitting key using a secure channel by both encrypting and decrypting. In this we use a currently emerging area of cryptography viz the uncertainty principle of cryptography. In conventional information theory and cryptography, the information or data fed is always monitored or can undergo threats easily as there is no safety. But by using QC for authentication, a quantum channel is established instead of an ordinary channel and it permits secure distribution of random keys or information between parties. This is the main advantage

of using QC. Message encryption is done using the code based algorithm. Cryptography is about the avoidance and recognition of fraud and other cruel activities. Symmetric-key cryptography, also called secret key cryptography. It involves the use of a secret key known only to the users. It is considered by the use of a single key to perform both the encrypting and decrypting of data. On October, 2, 2000, The National Institute of Standards and Technology (NIST) announced Rijndael as the new Advanced Encryption Standard (AES). The Predecessor to the AES was Data Encryption Standard (DES) which was considered to be insecure because of its weakness to brute force attacks. DES was a standard from 1977 and stayed until the mid1990's To overcome the situation, the National Institute of Standards and Technology (NIST) created a new encryption standard. The methods were proposed by Joan Daemon and Vincent Rijman, which are called Rijndael.

The National Institute of Standards and Technology (NIST) have published the specifications of this encryption standard in the Federal Information Processing Standards (FIPS) Publication 197. Different versions of AES algorithm exist today (AES128) depending on the size of the encryption key. Three architectural optimization approaches can be employed to speed up the hardware implementations: Pipelining, Sub Pipelining, and Loop-Unrolling. Among these approaches, the sub pipelined architecture can achieve maximum speed up and optimum speed-area ratio in non-feedback modes. The Rijndael

algorithm, the Advanced Encryption Standard (AES) provides a symmetric key cryptography that allows for the encryption and decryption of blocks of data. As a symmetric system, the secret key must be shared between the sender and receiver in order for communication to be possible.

AES algorithm is generally applied in the financial field in domestic, such as realizing legal encryption in ATM, magnetism card and intelligence card.

A cryptographic system uses 2 main blocks i.e. is encryption and decryption. Encryption is basically

2. EXPERIMENTAL SECTION

In our project we propose Triple AES algorithm using gray code cryptography. Triple AES is based on the AES steps but In an Encryption side we encrypt two times and decrypt one time to change data plain text to cipher text. In a decryption side, we decrypt two times and encrypt one time to change data cipher text in to plain text. In this system secret key is needed for encryption and decryption process. This secret key is generated using gray code key algorithm. Here since we are using gray concept, we use gray based encryption process to generate a gray coded sequence and this sequence is used as the key and is given to the Triple AES algorithm.

2.1.SOFTWARE DEVELOPMENT:

2.1.1GRAY CODE:

Code is a symbolic representation of discrete information. Codes are of different types. Gray Code is one of the most important codes. It is a non-weighted code which belongs to a class of codes called minimum change codes. In this codes while traversing from one step to another step only one bit in the code group changes. In case of Gray Code two adjacent code numbers differs from each other by only one bit. The idea of it can be cleared from the table given below.

As this code it is not applicable in any types of arithmetical

protecting information, i.e. data is transferred into unreadable form in order to ensure privacy and also keeps the information hidden. Decryption, being the reverse process of encryption uses the encrypted form of data and transforms this data into readable form. A key is transmitted between encryption and decryption modules which provide a secure channel for transferring information. This includes a basic block diagram of a cryptographic system, which has an encryption block a secret key and a decryption block, and these are the main components of a cryptographic system.

concentrate on the table of Gray Code given below where we can find the difference of binary code from gray code while traversing through the table for their respective decimal numbers.

Decimal numbers	Binary code	Gray code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

TABLE :BINARY TO GRAY CODE CONVERSION

From this table we can obtain the equivalent gray code of the decimal numbers. There are several steps which will make you understand how the codes are formed.
 (1) In case of gray code one bit will change from its previous in each steps. One thing must be kept in mind that the change of bit always occurs from the right side i.e from L.S.B towards the M.S.B. At first the first three bits are constant i.e 000 and the fourth bit changes from 0 to 1. We know that for binary digit possible combination is 0 and 1, so keeping first three bit constant the possible combination of 4th bit is over for decimal 0 and 1 respectively.
 (2)Now move to the next bit from L.S.B i.e 3rd bit, that changes from 0 to 1 which is the decimal equivalent for 2. Now one more combination is left for the fourth bit keeping the first three constant i.e 001. We can change 4th bit from 1 to 0. Thus the gray code for decimal number 3 is 0010.

- Pavithra.SAssistant Professor, Electronics and Communication engineering in Saveetha School of Engineering , India
- Vaishnavi.M ,Umadevi.V , Vinothini.M are currently pursuing Bachelor degree program in Electronics and Communication engineering in Saveetha School of Engineering , India, PH-09566257527. E-mail: vaishu.maheshwaran@gmail.com

operations but it has some applications in analog to digital converters and in some input/output devices. Now let us

(3) Traverse to the next code. Here we can do only one thing i.e we can change the second bit as all possible combinations are over. Question may strike in your mind that why can't we change the third bit again which will also be a one bit change from its previous. But changing third bit would give the equivalent gray code 0000 which has occurred earlier. So you must remember that a number occurring previously must not be repeated. So the equivalent code for 4 will be 0110. Here only the second bit has changed from the previous code. Now again we will keep first and second bit constant and find the possible combinations of the third and the fourth bit by only changing 1 bit in each steps.

Now for 5 only the fourth bit has changed. Again for 6 only the third bit is changed keeping others constant. Lastly at 7 again the fourth bit has changed from 1 to 0 where all other bits are constant. In 8 you can see that the equivalent gray code is 1100. Here the 1st bit changes from 0 to n1 as all the combination of the 2nd,3rd and 4th bits are completed keeping the 1st constant at 0. Now in same way the 1st bit is kept constant and all the possible combination changing single bit in each step from right to left is done.

Binary to gray code conversion is a very simple process. There are several steps to do this types of conversions. Steps given below elaborate on the idea on this type of conversion.

(1) The M.S.B. of the gray code will be exactly equal to the first bit of the given binary number.

(2) Now the second bit of the code will be exclusive-or of the first and second bit of the given binary number, i.e if both the bits are same the result will be 0 and if they are different the result will be 1.

(3) The third bit of gray code will be equal to the exclusive-or of the second and third bit of the given binary number. Thus the Binary to gray code conversion goes on.

2.1.3 AES:

AES is an iterated block cipher with a variable block length and a variable key length. The block length and the key length can be independently specified to 128 bits. Cipher Key and the number of rounds The Cipher Key is similarly pictured as a rectangular array with four rows. The number of columns of the Cipher Key is denoted by N_k and is equal to the key length divided by 32. These representations are illustrated in Figure .In some instances, these blocks are also considered as one-dimensional arrays of 4-byte vectors, where each vector consists of the corresponding column in the rectangular array representation. These arrays hence have lengths of 4, 6 or 8 respectively and indices in the ranges 0..3, 0..5 or 0..7. 4-byte vectors will sometimes be referred to as words. Where it is necessary to specify the four individual bytes within a 4-byte vector or word the notation (a, b, c, d) will be used where a, b, c and d are the bytes at positions 0, 1, 2 and 3 respectively within the column, vector or word being

considered.

The input and output used by Rijndael at its external interface are considered to be one dimensional arrays of 8-bit bytes numbered upwards from 0 to the $4 \cdot N_b - 1$. These blocks hence have lengths of 16, 24 or 32 bytes and array indices in the ranges 0..15, 0..23 or 0..31. The Cipher Key is considered to be a one-dimensional arrays of 8-bit bytes numbered upwards from 0 to the $4 \cdot N_k - 1$. These blocks hence have lengths of 16, 24 or 32 bytes and array indices in the ranges 0..15, 0..23 or 0..31.

The cipher input bytes (the "plaintext" if the mode of use is ECB encryption) are mapped onto the state bytes in the order $a_{0,0}, a_{1,0}, a_{2,0}, a_{3,0}, a_{0,1}, a_{1,1}, a_{2,1}, a_{3,1}, a_{4,1} \dots$, and the bytes of the Cipher Key are mapped onto the array in the order $k_{0,0}, k_{1,0}, k_{2,0}, k_{3,0}, k_{0,1}, k_{1,1}, k_{2,1}, k_{3,1}, k_{4,1} \dots$. At the end of the cipher operation, the cipher output is extracted from the state by taking the state bytes in the same order. Hence if the one-dimensional index of a byte within a block is n and the two dimensional index is (i, j) , we have:

$$i = n \bmod 4 ; j = \lceil n / 4 \rceil ; n = i + 4 \cdot j$$

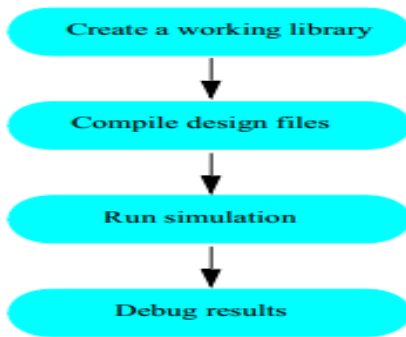
Moreover, the index i is also the byte number within a 4-byte vector or word and j is the index for the vector or word within the enclosing block. The number of rounds is denoted by N_r and depends on the values N_b and N_k . It is given in Table.

N_r	$N_b = 4$	$N_b = 6$	$N_b = 8$
$N_k = 4$	10	12	14
$N_k = 6$	12	12	14
$N_k = 8$	14	14	14

TABLE : NUMBER OF ROUNDS (N_r) AS A FUNCTION OF THE BLOCK AND KEY LENGTH.

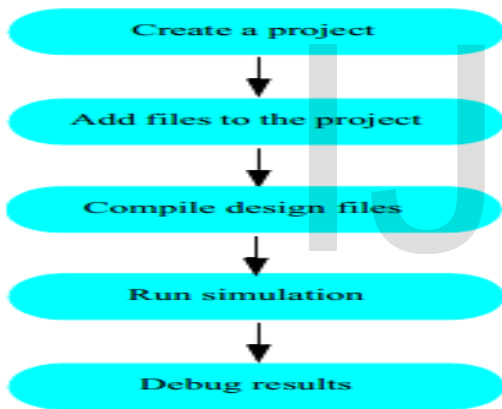
2.1.3 ModelSim

ModelSim is a very powerful simulation environment, and as such can be difficult to master. Thankfully with the advent of Xilinx Project Navigator 6.2i, the Xilinx tools can take care of launching ModelSim to simulate most projects. However, a rather large flaw in Xilinx Project Navigator 6.2i is its inability to correctly handle test benches which instantiate multiple modules. To correctly simulate a test bench which instantiates multiple modules, you will need to create and use a ModelSim project manually. Model Sim is a simulation and debugging tool for VHDL, Verilog, and mixed-language designs. **Basic simulation flow**



Project flow

A project is a collection mechanism for an HDL design under specification or test. Even though you don't have to use projects in ModelSim, they may ease interaction with the tool and are useful for organizing files and specifying simulation settings. The following diagram shows the basic steps for simulating a design within a ModelSim project



2.1.4 VERILOG

Verilog, standardized as IEEE 1364, is a hardware description language (HDL) used to model electronic systems. It is most commonly used in the design and verification of digital circuits at the register-transfer level of abstraction. It is also used in the verification of analog circuits and mixed-signal circuits.

Verilog HDL is one of the two most common Hardware Description Languages (HDL) used by integrated circuit (IC) designers. The other one is VHDL. HDL's allows the design to be simulated earlier in the design cycle in order to correct errors or experiment with different architectures. Designs described in HDL are technology-independent,

easy to design and debug, and are usually more readable than schematics, particularly for large circuits.

Verilog can be used to describe designs at four levels of abstraction:

- (i) Algorithmic level (much like c code with if, case and loop statements).
- (ii) Register transfer level (RTL uses registers connected by Boolean equations).
- (iii) Gate level (interconnected AND, NOR etc.).
- (iv) Switch level (the switches are MOS transistors inside gates).

3. HARDWARE DEVELOPMENT:

AVR1318: Using the XMEGA built-in AES accelerator :

The XMEGA™ AES Crypto Module supports the Advanced Encryption Standard (AES), and can perform encryption and decryption. The module supports a key length of 128 bits. The 128-bit key block and 128-bit data block (plaintext or ciphertext) must be loaded into the Key and State memory in the AES Crypto Module. The AES uses 375 clock cycles to execute one encryption/decryption after the Key and State memory is loaded and the mode of operation is selected.

3.1 ENCRYPTION:

To execute an AES encryption using the AES Crypto Module the following should be done.

- Enable/disable AES interrupts, by setting/clearing the Interrupt priority and enable bits (INTLVL) in the Interrupt Control register (INTCTRL).
- Select the AES encryption direction, by clearing the decrypt bit (decrypt) in the control register (CTRL).
- Load the AES key into the AES Key memory
- Load the data block into the AES State memory

- Start encryption, by setting the start bit (START) in the control register (CTRL).

When the encryption is completed, the AES State Ready Interrupt Flag (SRIF) in the AES Status register (STATUS) is set. If the interrupt mechanism is used an interrupt is generated. The AES State memory will after an encryption is completed contain the generated ciphertext while the AES Key memory will contain the last subkey of the Expanded Key defined in the AES standard.

3.2 DECRYPTION

To execute an AES decryption using the AES Crypto Module the following should be done.

- Enable/disable AES interrupts, by setting/clearing the Interrupt priority and enable bits (INTLVL) in the Interrupt Control register (INTCTRL).
- Select the AES decryption direction, by setting the decrypt bit (decrypt) in the control register (CTRL).
- Load the last subkey of the Expanded Key defined in the AES standard into the AES Key memory.
- Load the data block into the AES State memory
- Start decryption, by setting the start bit (START) in the control register (CTRL).

When the decryption is completed, the AES State Ready Interrupt Flag (SRIF) in the AES Status register (STATUS) is set. If the interrupt mechanism is used an interrupt is generated. The AES State memory will after a decryption is completed contain the generated plaintext while the AES Key memory will contain the original Key defined in the AES standard.

4. RESULTS AND DISCUSSION :

4.1 SOFTWARE:

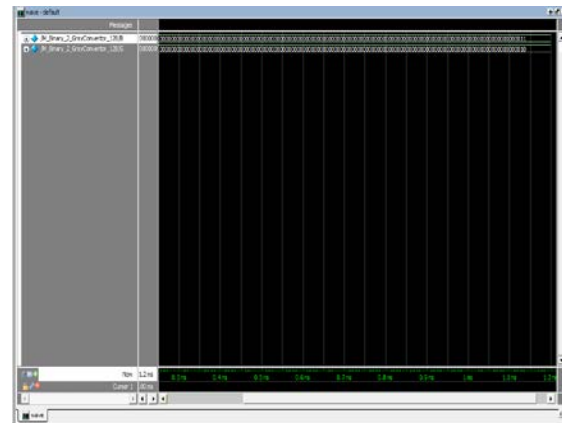


Fig. Binary key encode to gray code

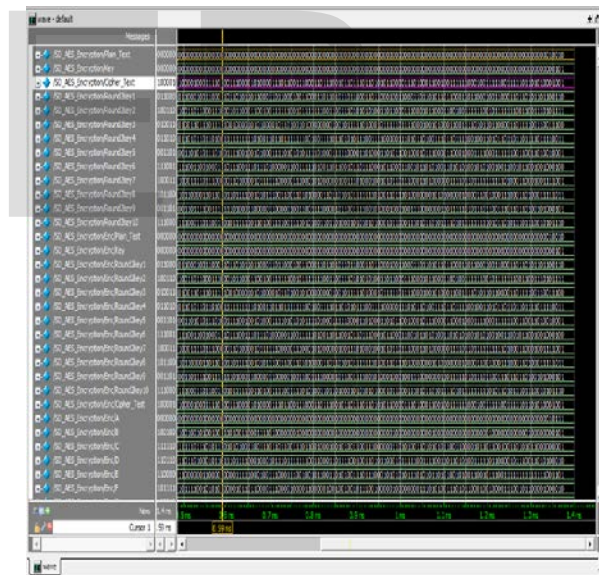


Fig. AES encryption

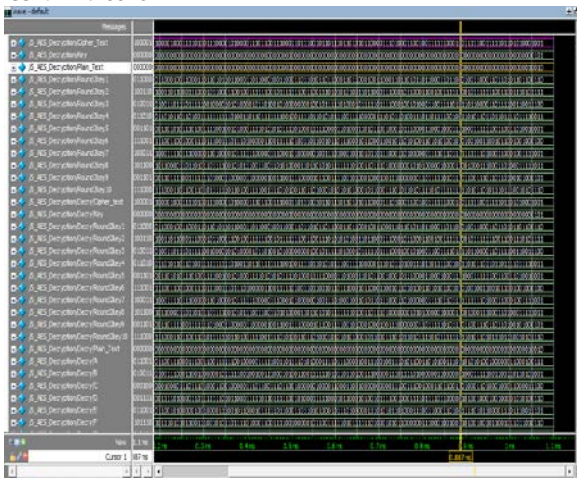


Fig. AES decryption

Name	Value	Type
data	{uint8_t[16](data@0x2050)}	uint8_t[16]
[0]	1	uint8_t
[1]	2	uint8_t
[2]	3	uint8_t
[3]	4	uint8_t
[4]	5	uint8_t
[5]	6	uint8_t
[6]	7	uint8_t
[7]	8	uint8_t
[8]	9	uint8_t
[9]	16	uint8_t
[10]	17	uint8_t
[11]	18	uint8_t
[12]	19	uint8_t
[13]	20	uint8_t
[14]	21	uint8_t
[15]	22	uint8_t

Fig. plain text

```

/* Do AES encryption and decryption of a single block. */
success = AES_encrypt(data, single_ans, key);
success = AES_decrypt(single_ans, single_ans, lastsubkey);
    
```

Name	Value	Type
single_ans	{uint8_t[16](data@0x20d1)}	uint8_t[16]
[0]	181	uint8_t
[1]	158	uint8_t
[2]	216	uint8_t
[3]	166	uint8_t
[4]	15	uint8_t
[5]	237	uint8_t
[6]	245	uint8_t
[7]	130	uint8_t
[8]	11	uint8_t
[9]	193	uint8_t
[10]	186	uint8_t
[11]	163	uint8_t
[12]	195	uint8_t
[13]	238	uint8_t
[14]	157	uint8_t
[15]	233	uint8_t

Fig. Encryption

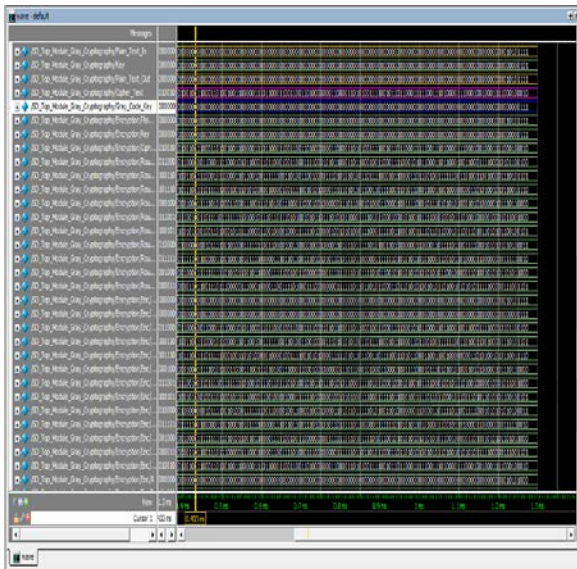


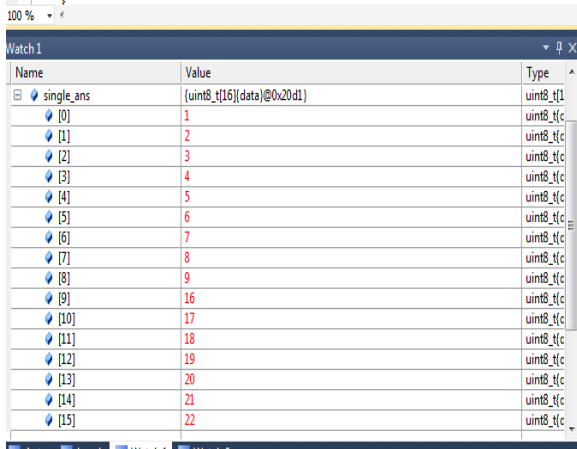
Fig. AES operation on Gray code key

4.2HARDWARE:

```

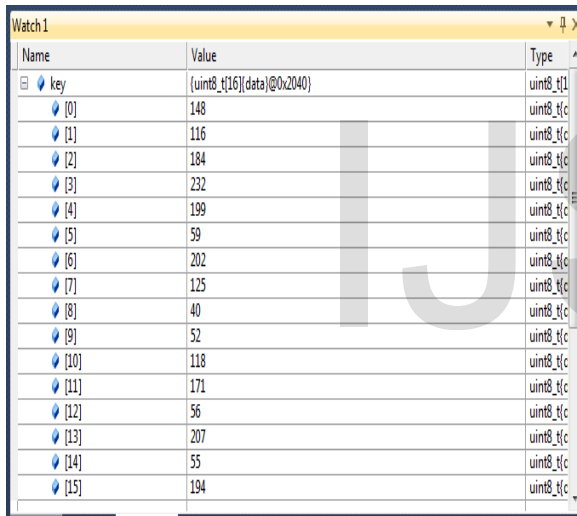
success = AES_decrypt(single_ans, single_ans, lastsubkey);

/* Check if decrypted answer is equal to plaintext. */
for(uint8_t i = 0; i < BLOCK_LENGTH; i++){
    if (data[i] != single_ans[i]){
        success = false;
    }
}
    
```



Name	Value	Type
single_ans	{uint8_t[16](data)@0x20d1}	uint8_t[16]
[0]	1	uint8_t
[1]	2	uint8_t
[2]	3	uint8_t
[3]	4	uint8_t
[4]	5	uint8_t
[5]	6	uint8_t
[6]	7	uint8_t
[7]	8	uint8_t
[8]	9	uint8_t
[9]	16	uint8_t
[10]	17	uint8_t
[11]	18	uint8_t
[12]	19	uint8_t
[13]	20	uint8_t
[14]	21	uint8_t
[15]	22	uint8_t

Fig. Decryption



Name	Value	Type
key	{uint8_t[16](data)@0x2040}	uint8_t[16]
[0]	148	uint8_t
[1]	116	uint8_t
[2]	184	uint8_t
[3]	232	uint8_t
[4]	199	uint8_t
[5]	59	uint8_t
[6]	202	uint8_t
[7]	125	uint8_t
[8]	40	uint8_t
[9]	52	uint8_t
[10]	118	uint8_t
[11]	171	uint8_t
[12]	56	uint8_t
[13]	207	uint8_t
[14]	55	uint8_t
[15]	194	uint8_t

Fig. Key

5.ADVANTAGES

- Triple AES is more secure (it is less susceptible to cryptanalysis than 3DES).
- Triple AES supports larger key sizes than 3DES's 112 bytes.
- Triple AES is faster in both hardware and software.
- Triple AES's 128-bit block size makes it less open to attacks via the birthday problem than 3DES with its 64-bit block size.

- AES is required by the latest U.S. and international standards.
- High efficiency of gray code key
- High reliability

6. CONCLUSION

The triple AES algorithm was chosen as the new Advanced Encryption Standard (AES) for several reasons. The purpose was to create an algorithm that was resistant against known attacks, simple, and quick to code. Choosing to use field GF(2) 8 was a very good decision.. In fact, every operation is invertible by design. In addition, the block size and key size can vary making the algorithm versatile. AES was originally designed for non-classified U.S. government information, but, due to its success. This project gives a concise outline about cryptography and gray code based cryptography. Information about technologies used in gray code is also provided here. It also discusses about secure message transfer between two systems. The proposed system is computationally more efficient. They provide best security and are faster to execute.

7. REFERENCES

1. Table of general binary codes. An updated version of the tables of bounds for small general binary codes given in M.R. Best, A.E. Brouwer, F.J. MacWilliams, A.M. Odlyzko& N.J.A. Sloane (1978), "Bounds for Binary Codes of Length Less than 25", IEEE Trans. Inf. Th. 24: 81-93.
2. Press, WH; Teukolsky, SA; Vetterling, WT; Flannery, BP (2007). "Section 22.3.Gray Codes". Numerical Recipes: The Art of Scientific Computing (3rd Ed.). New York: Cambridge University Press. ISBN 978-0- 521-88068-8.
3. Savage, Carla (1997). "A Survey of Combinatorial Gray Codes". SIAM Rev. 39 (4):605629doi:10.1137/S003614459529527 2. JSTOR 2132693.
4. N.C. Seiler. Gray code DAC ladder, US Patent 4,591,826, 1986.