International Journal of Science and Engineering Applications
Volume 6 Issue 03, 2017, ISSN-2319-7560 (Online)

# Implementation of Advanced Encryption Standard Algorithm with Key Length of 256 Bits for Preventing Data Loss in an Organization

Owusu Nyarko- Boateng
Innerjoy Digital Systems
Sunyani, Ghana

Michael Asante
Department of Computer Science
Kwame Nkrumah University of Science and Technology,
Kumasi, Ghana

Isaac Kofi Nti
Department of Electrical/Electronic Engineering
Sunyani Technical University
Sunyani, Ghana

**Abstract**: Data and network security is one among the foremost necessary factors in today's business world. In recent, businesses and firms like financial institutions, law firms, schools, health sectors, telecommunications, mining and a number of government agencies want a strategic security technique of managing its data. Organizations managing bigger monetary information, bio-data and alternative relevant info are losing its valuable information or data at rest, in usage or in motion to unauthorized parties or competitors as a result of activities of hackers. Organizations are losing millions of dollars as a result of unprotected data that gets into the hands of malicious persons [1]. Data protection in an organization has become vital in today's business. In order to possess secure information, this information should be protected in order that although malicious persons get access to the info, it becomes wealth-less and useless to them. Advanced Encryption Standard (AES), could be a scientific discipline rule that may be used for secured data and communication in an organization, it uses same key that's isobilateral key for transmission additionally as reception. The AES rule is capable of using cryptographic keys of 128, 192, and 256 bits, this paper implement AES block cipher of 256-bits and 256-bit key size, developed with C# as a front-end client machine and MS SQL used for the database as a back-end machine.

**Keywords**: Cryptography; AES-algorithm; Decryption; Encryption; Cipher; 256-bit-key

## 1. INTRODUCTION

The scrambling of plaintext delivers a safe and nice significance for secured data and communication. Use of scientific discipline algorithms is completed for the aim of security in varied applications like secured optical disk content, ATM, etc. Secured data and communication in an organization is one in all the foremost necessary things in present day business and its requirement is rapidly increasing [2] [3]. With the introduction of LAN, WAN, MAN and internet technology in recent years, the computer network communication is exposed to unwanted people giving them access to pose different kinds of attacks on personal and organizational data in a network environment [4]. Every individual desires, their information to be secured and privacy should be maintained. This demand is consummated by the employment of cryptography. Numerous security systems are needed to guard a shared information in an organization. The paper focuses on cryptography to secure the info of an organization in motion within its network, at rest, and in usage to unauthorized parties.

### 1.1 Cryptography

In information, communications and networking, cryptography is important once human action over a transmission medium, particularly the un-trusted medium, significantly the net [5]. The encrypted knowledge is named as ciphertext. At the receiver part, solely those that have a secret key can decipher the message into plain text to obtaining the initial knowledge [6]. Generally encrypted messages are often lessened by cryptography, which is thought as code breaking. Cryptography is often classified into 2 varieties symmetric-key systems and Asymmetric-key systems [3]. In symmetric-key secret writing systems sender and receiver of the message, create use of the identical key, this distinctive secret is used for secret writing also as coding of the message [3]. In all cases as illustrated in Figure 1, the original unencrypted data is referred to as plaintext (X). It is encrypted into ciphertext (Y), which will in turn (usually) be decrypted into usable plaintext with the same key (K) used for encryption.

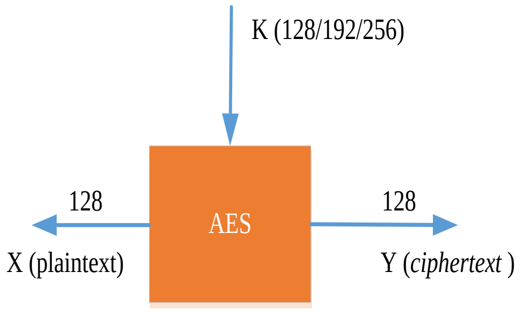


**Figure 1 Cryptography Process**

Cryptography does not only protect data from theft or alteration, but can also be used for user authentication in most application [5].

## 2. DESCRIPTION OF AES ALGORITHM

AES is a block cipher which is also known as Rijndael algorithm and it was developed by Joan Daemen and Vincent Rijmen. The algorithm can efficiently be executed on a variety of computer processors and hardware's [7]. AES relies on a style principle referred to as a substitution-permutation network, a mix of each substitution and permutation, and is quick in each software system and hardware. In contrast to its forerunner DES, AES doesn't use a Feistel network [7]. The robust AES development process and its complex internal structures ensures very secure algorithm and has no known weaknesses. In accordance with the AES requirements, Rijndael's key length can be 128, 192 or 256-bits. Rijndael algorithm is made up of variable block size that can also be 128, 192, or 256-bits. This implies that, a Rijndael algorithm with key sizes of 128, 192 and 256-bits provides approximately [8]. AES is one of the most up-to-date out of the four current algorithms approved for federal United States within the US [9]. AES operates on a four × four (4 x4) column-major order matrix of bytes, termed the state, though some versions of Rijndael have a bigger block size and have further columns within the state.

### 2.1 AES internal structure

Figure 2 shows the overall structure of the AES algorithm, AES composition and building blocks was designed based on standard known as a substitution-transformation arrangement with fixed block size of 128 bits, and a key size of 128, 192, or 256 bits and has a high-speed in both software and hardware. Unlike its predecessor DES, AES is not based on Feistel network [5]. The principle of AES design is known as a substitution-permutation network, which is the combination of both substitution and permutation. AES operates on a 4 × 4 column-major order matrix of bytes, known as the state. However, some versions of Rijndael have a larger block size and have additional columns in the state. Generally, AES calculations are done in a special finite field called Galois Fields, which allows mathematical operations to scramble data easily and effectively [10] [11]. There are numerous rounds within the AES encryption development. Each operational round consists of more than a few processing steps, each one containing four similar but different stages, including the one that depends on the encryption key itself. The various stages are;

- ✓ ByteSub
- ✓ ShiftRows
- ✓ MixColumn
- ✓ addRoundkey

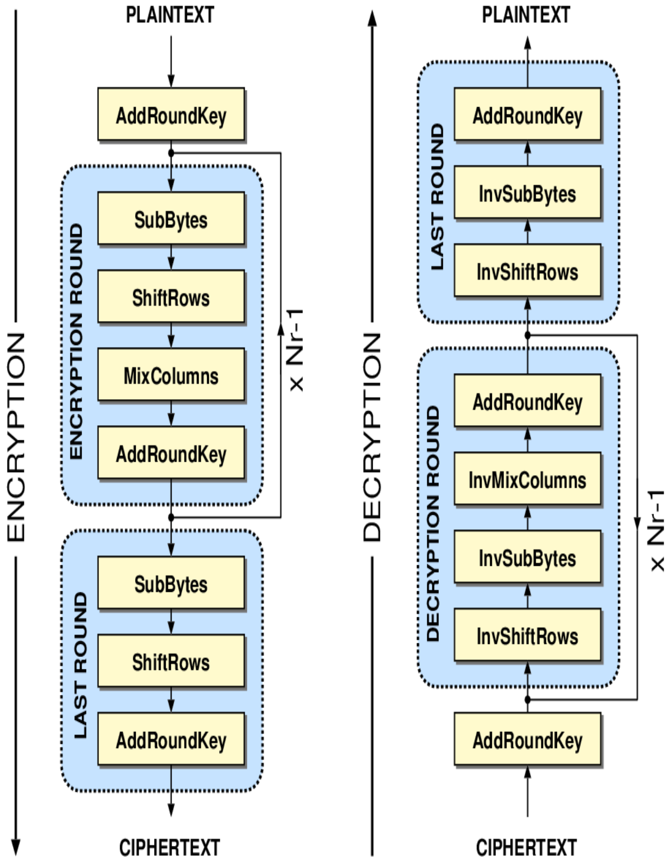


**Figure 2 Overall Structure of AES Algorithm (Source: http://crypto.stackexchange.com/questions/8043/aes-addroundkey)**

### 2.2 Encryption

The preliminary key is added to the input value at the beginning stage in the encryption mode, which is called a preliminary round. Several repetitions follows immediately after the initial round with a slightly modification of the final. The following operation are perform in every one round respectively.

#### *2.2.1 SubBytes Conversion*

The SubBytes convention stage is a non-linear byte replacement, where each state byte is operated independently. This is achieved through an S-box. S-box is a pre-calculated replacement table which holds 256 numbers (from 0 - 255) and their matching resulting value. The SubBytes step has each byte in ai,j in the state matrix swapped with a SubByte bi,j by means of an 8 bits substitution box, known as Rijndael S-box. However, the SybBytes transform algorithm is based on Galois Field Inverse operation GF (28) known to have excellent non-linearity properties. The use of Galois Field is to prevent attacks based on simple algebraic properties. The S-box is created by merging the inverse function with an invertible affine transformation. The S-box is selected in order to prevent any fixed operational networks, thus, ai,j ≠ bi,j. The S-box is created by determining the multiplicative inverse for a given number in GF (28) = GF (2) [x]/($x^8 + x^4 + x^3 + x + 1$), Rijndael's finite field. Zero, that has no inverse, is mapped to zero. The multiplicative opposite is then changed using the following affine transformation [12]

$$\begin{bmatrix}1&0&0&0&1&1&1&1\\1&1&0&0&0&1&1&1\\1&1&1&0&0&0&1&1\\1&1&1&1&0&0&0&1\\1&1&1&1&1&0&0&0\\0&1&1&1&1&1&0&0\\0&0&1&1&1&1&1&0\\0&0&0&1&1&1&1&1\end{bmatrix}\begin{bmatrix}x_0\\x_1\\x_2\\x_3\\x_4\\x_5\\x_6\\x_7\end{bmatrix}+\begin{bmatrix}1\\1\\0\\0\\0\\1\\1\\0\end{bmatrix}$$

Where [x7, ..., x0] is the multiplicative inverse as a vector. This transformation is that the total of multiple rotations of the byte as a vector, wherever addition is that the XOR operation. The matrix development can be deduced using the following algorithm:

Let input number be (s) (thus an unsigned 8 bit variable).

Let the values result be 0.

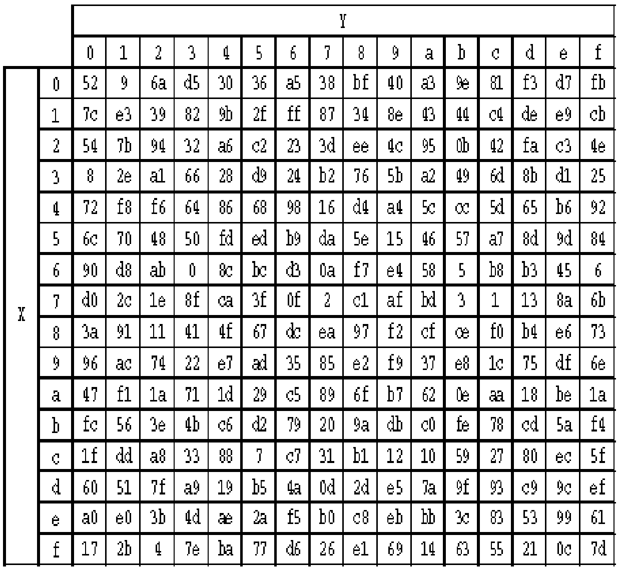
For 5 times:

XOR result with s.

Revolve s one bit to the left.

Once the matrix exponentiation is done, XOR the value by the decimal number 99 (the hexadecimal number 0x63, the binary number 0b01100011, the bit string 11000110 demonstrating the number in LSb first notation) [12]. This steps will produce an S-box table as shown in Table 1.

**Table 1 S-BOX TABLE**



| | | Y | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
| X | 0 | 52 | 9 | 6a | d5 | 30 | 36 | a5 | 38 | bf | 40 | a3 | 9e | 81 | f3 | d7 | fb |
| | 1 | 7c | e3 | 39 | 82 | 9b | 2f | ff | 87 | 34 | 8e | 43 | 44 | c4 | de | e9 | cb |
| | 2 | 54 | 7b | 94 | 32 | a6 | c2 | 23 | 3d | ee | 4c | 95 | 0b | 42 | fa | c3 | 4e |
| | 3 | 8 | 2e | a1 | 66 | 28 | d9 | 24 | b2 | 76 | 5b | a2 | 49 | 6d | 8b | d1 | 25 |
| | 4 | 72 | f8 | f6 | 64 | 86 | 68 | 98 | 16 | d4 | a4 | 5c | cc | 5d | 65 | b6 | 92 |
| | 5 | 6c | 70 | 48 | 50 | fd | ed | b9 | da | 5e | 15 | 46 | 57 | a7 | 8d | 9d | 84 |
| | 6 | 90 | d8 | ab | 0 | 8c | bc | d3 | 0a | f7 | e4 | 58 | 5 | b8 | b3 | 45 | 6 |
| | 7 | d0 | 2c | 1e | 8f | ca | 3f | 0f | 2 | c1 | af | bd | 3 | 1 | 13 | 8a | 6b |
| | 8 | 3a | 91 | 11 | 41 | 4f | 67 | dc | ea | 97 | f2 | cf | ce | f0 | b4 | e6 | 73 |
| | 9 | 96 | ac | 74 | 22 | e7 | ad | 35 | 85 | e2 | f9 | 37 | e8 | 1c | 75 | df | 6e |
| | a | 47 | f1 | 1a | 71 | 1d | 29 | c5 | 89 | 6f | b7 | 62 | 0e | aa | 18 | be | 1a |
| | b | fc | 56 | 3e | 4b | c6 | d2 | 79 | 20 | 9a | db | c0 | fe | 78 | cd | 5a | f4 |
| | c | 1f | dd | a8 | 33 | 88 | 7 | c7 | 31 | b1 | 12 | 10 | 59 | 27 | 80 | ec | 5f |
| | d | 60 | 51 | 7f | a9 | 19 | b5 | 4a | 0d | 2d | e5 | 7a | 9f | 93 | c9 | 9c | ef |
| | e | a0 | e0 | 3b | 4d | ae | 2a | f5 | b0 | c8 | eb | bb | 3c | 83 | 53 | 99 | 61 |
| | f | 17 | 2b | 4 | 7e | ba | 77 | d6 | 26 | e1 | 69 | 14 | 63 | 55 | 21 | 0c | 7d |

### 2.2.2 *ShiftRows*

In ShiftRows transformation stage, each row of the state is intermittently left shifted above diverse offsets. Row 0 remain in position, whiles row 1, 2 and 3 are shifted one byte, two bytes and three bytes to the left respectively [13].

### 2.2.3 *MixColumn*

From the MixColumn operations, there is a transposition of linear transformation made to join the 4-byte in each column as shown in fig 2.4. The task of this step is to take 4-byte as input and outputs 4-byte, where every input bytes have an effect on all the output 4-byte. Each column is transformed using fixed matrix operations; this is composed of multiplication and addition of the entries as illustrated in fig 2.5. Addition is simply XOR. Multiplication is modulo irreducible polynomial [14] [15]. In the MixColumn process, each column is treated as a polynomial over GF (28) and is then multiplied modulo with a fixed matrix polynomial c(x) multiplies every column, thus,

C (x) = $3x^3 + x^2 + x + 2$

$$\begin{bmatrix}s'_{0,c}\\s'_{1,c}\\s'_{2,c}\\s'_{3,c}\end{bmatrix}=\begin{bmatrix}02&03&01&01\\01&02&03&01\\01&01&02&03\\03&01&01&02\end{bmatrix}\begin{bmatrix}s_{0,c}\\s_{1,c}\\s_{2,c}\\s_{3,c}\end{bmatrix}$$

$(0 \le c < N_b)$

### 2.2.4 *addRoundkey*

For every round in the AddRoundKey step, a subkey is generated from the main key by means of Rijndael's key schedule. The subkey is combined with the state, and that notwithstanding each subkey is the same size as the state. The subkey is inserted by combining every byte in the state with it related byte in the subkey by means of bitwise XOR [16].

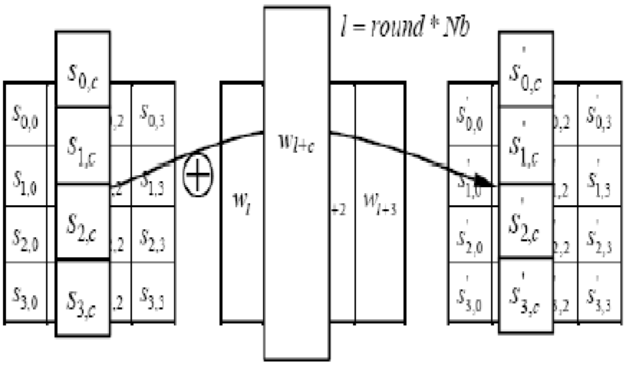


**Figure 3 AddRoundKey XORs each column of the State with a word from the key schedule.**

## 2.3 RELATED WORKS

A FPGA-based implementation of the Advanced Encryption Standard (AES) algorithm was proposed by [16]. The design employs an iterative looping technique with block and a 128 bits key size S-box table implementation. The research concluded that low complexity architecture and easily achieves low latency as well as high throughput 1054Mbit/sec for encryption and 615 Mbit/sec for decryption was achieved.

An implementation of high speed AES algorithm with Key Length of 256 Bits based on FPGA is presented was proposed by [17] to advance the security of data in motion. [18] proposed FPGA-based implementation of the Advanced Encryption Standard (AES) algorithm in UART Module. The design employs an iterative looping technique with block and a 128 bits key size S-box table implementation. A throughput 1054Mbit/sec for encryption and 615 Mbit/sec for decryption was achieved in their research. A proposed FPGA-based implementation of the Advanced Encryption Standard (AES) algorithm was also presented by [15]. The design employs an iterative looping technique with block and a 128 bits key size S-box table implementation. Their design was executed using APEX20KC FPGA on Altera which is based on great performance design. In their paper, a low latency and the throughput attained a value of 1054Mbit/sec for encryption and 615 Mbit/sec for decryption [15]. An implementation of AES encryption and decryption standard AES-128 was proposed by [19]. All the transformations of each secret writing Associate in nursing decoding are simulated victimization an unvarying style approach so as to reduce the hardware consumption. Their paper proposed that their methodology will create it a really low-complex design, particularly in saving the hardware resource in implementing the AES InverseSub Bytes module and Inverse combine columns module. Because the S - box is enforced by look-up-table during this style, the chip space and power will still be optimized. The new combine Column transformation improves the performance of the inverse cipher and additionally reduces the quality of the system that supports the inverse cipher. As a result this transformation has comparatively low relevant diffusion power .This allows for scaling of the design towards vulnerable moveable and cost-sensitive communications devices in client and military applications [19]. An implementation of the 128 bit AES on a Field Programmable Gate Array (FPGA) was proposed for significant level of security similarly as quicker time interval in order that it will be used for secured communication of ATM, optical disc content similarly as for secured storage of confidential company documents, government documents [3].

The above discussions show that an implementation of AES algorithm with 128 key length can be enhanced with a 256 bits key length to provide a well secured data [9]. Rijndael's reduced variety of rounds for smaller keys offers it a speed advantage, however it additionally reduces the security issue for those key sizes [7], hence this paper focuses on the implementation of the AES algorithm with a 256 bits key length to prevent data loss in an organization.

## 3. TOOLS AND METHODS

Microsoft Visual Studio 2012 (C#) was employed to develop the face, where system users (client) will diagrammatically communication to the server via a web browser. The rear finish (database) was developed with Microsoft Structured command language (MSSQL) server 2008. A Wireshark hacking tool was used for testing the encrypted data.

### 3.1 Design concept

Figure 4, shows the pictorial view of the proposed layout for the implementation of the AES algorithm with 256bits key length for organizational data protection. Records entered by the organization system users from the client machine from remote stations are encrypted using the ASE 256bit algorithm on the remote station and sent or transmitted to the data centre (server) true the clouds (internet) or LAN and MAN.

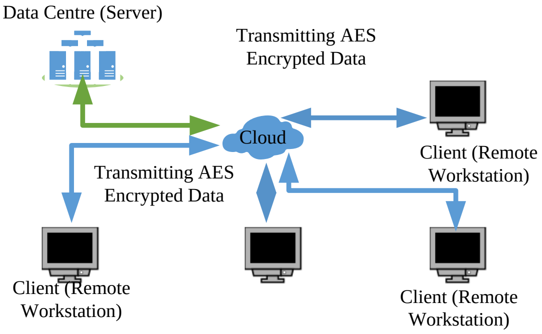
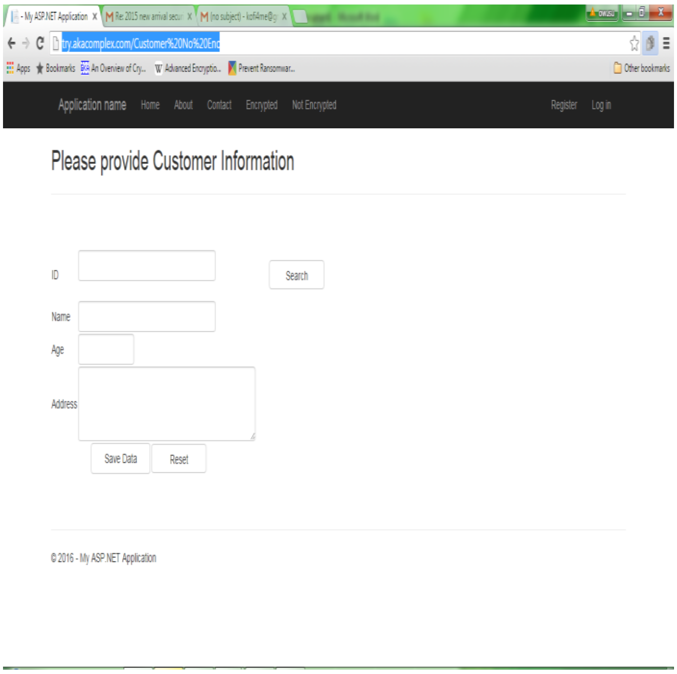

Data Centre (Server)
Transmitting AES Encrypted Data
Cloud
Client (Remote Workstation)
Transmitting AES Encrypted Data
Client (Remote Workstation)
Client (Remote Workstation)

**Figure 4 Conceptual Design of Proposed System**

For implementation, a web application with login details of client as shown in figure 5 was developed to collect raw data in plaintext to be encrypted by the AES for testing the outcome of the implementation of the AES 256bits algorithm.



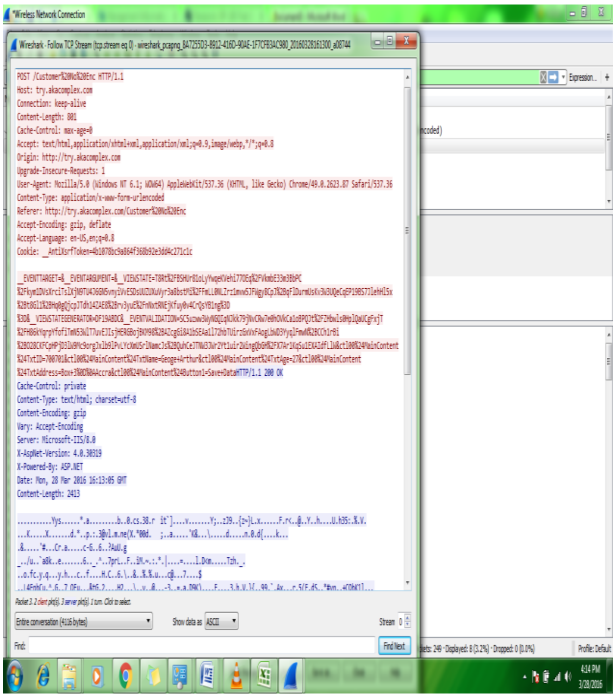
Please provide Customer Information

© 2016 - My ASP.NET Application

**Figure 5 Plaintext data collection interface**

## 4. TEST RESULTS AND DISCUSSIONS

To see the effectiveness of our AES 256bits implementation, five (5) different data was collect from the front end. The data transmission from the client machine to server was done in two scenario A and B. In scenario A the data was transmitted raw (no encryption) over vulnerable medium to the server. The Wireshark hacking tool was used to intercept the data packets in transmission and the raw format of the packets are displayed by the network packet analyzer in a plain text as illustrated in figure 6. This shows transmission on unencrypted data over a network is exposed to fallen into the hands of unauthorized persons.



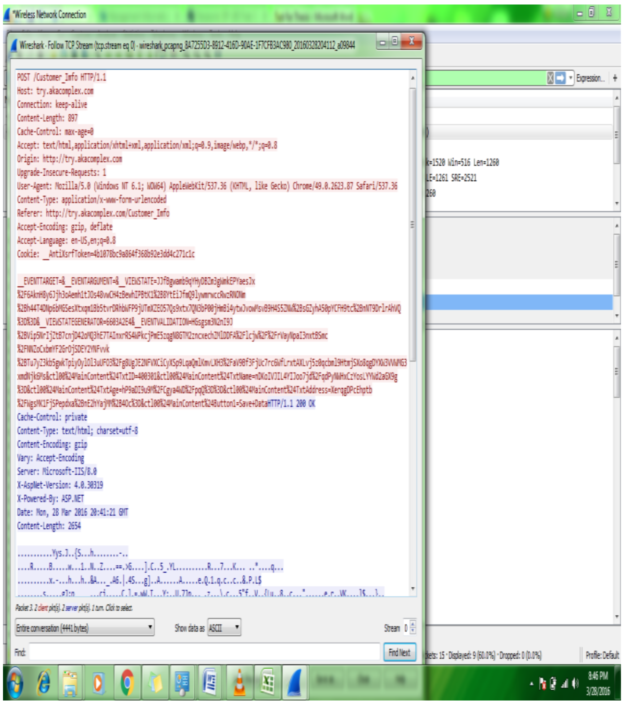
*Wireless Network Connection
Wireshark · Follow TCP Stream (tcp.stream eq 0) · wireshark_pcapng_8A7255D3-8912-416D-90AE-1F7CFB3AC980_20160328161300_a08744

```
POST /Customer%20No%20Enc HTTP/1.1
Host: try.akacomplex.com
Connection: keep-alive
Content-Length: 801
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Origin: http://try.akacomplex.com
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.87 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Referer: http://try.akacomplex.com/Customer%20No%20Enc
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8
Cookie: __AntiXsrfToken=4b1078bc9a864f368b92e3dd4c271c1c

__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=T8Rt%2FBSHUr81oLyYwqeKVehi77OEq%2FVkmbE33m3BbPC
%2Fkym1DVsXrciTslXjN9TU4J6GM5vnyiVvESDsUUZUXuVyr3a8bstMi%2FFmLL0NLZrz1mww5JFWgy8CpJ%2BqFlDurmUsKv3W3UQeCqEP19B57JlehHl5x
%20t8Gl1%20Hq0gQjcpJTdh14ZAE8%2Brv3yuE%2FnNxtRNEjKfuy0v4CrQsYB1ng%3D
%3D&__VIEWSTATEGENERATOR=DF19ABDC&__EVENTVALIDATION=SC5uzww5WyN6QIqMJkk79jNvCRw7e0hOVkCa1o8PQJt%2FZHbwls0HplQaUCgFxjT
%2FH86kYqrpYfofiTmN53WlT7uvEJIsjHER6BojBKM98%2BAZcg6i8A1bSEAa1l72hbTUirzGxVxFAogLbWD3YyqlFmwM%2BCCh1rBi
%2BO28CKFCpHPjD3lW9Mc9orgJxlb9lPvLYcXmUSrlNamcJs%2BQuhCeJTNW3JWr2Yt1uir2WingQbGH%2FX7Ar1KqSu1EXAIdfLlW&ctl00%24MainContent
%24TxtID=700701&ctl00%24MainContent%24TxtName=Geoge+Arthur&ctl00%24MainContent%24TxtAge=27&ctl00%24MainContent
%24TxtAddress=Box+3%0D%0AAccra&ctl00%24MainContent%24Button1=Save+DataHTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=utf-8
Content-Encoding: gzip
Vary: Accept-Encoding
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Mon, 28 Mar 2016 16:13:05 GMT
Content-Length: 2413

...........Yys......*.a.........b..0.cs.38.r it`]....v.......Y;..zJ9..{z~}L.x......F.r<..@..Y..h....U.h35:.%.V.
...K.....X.......d.*..p.:.3@vl.m.ne(X.*00d.  ;..a.....'K&...\.....d.....n.0.d{....k...
.&.....'#...Cr.a.....c-6..6..?AuU.g
_../u..`a8k..e.......6.._.^..7prL..F..iM.~.:.*.|....=....l.Dkm.....Tzh._.
..o.fc.y.q...y.h...c..f....H.C..6.\..&..%.%.u...c@...7....$
```

Packet 3. 2 client pkt(s), 3 server pkt(s), 1 turn. Click to select.
Entire conversation (4116 bytes)
Show data as ASCII
Stream 0
Find:
Find Next

**Figure 6 Unencrypted data of the client with ID: 700701**

Same client with ID 700701 data was transmitting under scenario B, which fully encrypted with AES 256 bits block cipher before transmission. Again the Wireshark hacking tool was used to intercept the packets in transit but the network packet analyzer displayed scrambled letters which has no meaning to the hacker as shown in figure 7.



*Wireless Network Connection
Wireshark · Follow TCP Stream (tcp.stream eq 0) · wireshark_pcapng_8A7255D3-8912-416D-904E-1F7CFB3AC980_20160328204112_a09844

```
POST /Customer_Info HTTP/1.1
Host: try.akacomplex.com
Connection: keep-alive
Content-Length: 897
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Origin: http://try.akacomplex.com
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.87 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Referer: http://try.akacomplex.com/Customer_Info
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8
Cookie: __AntiXsrfToken=4b1078bc9a864f368b92e3dd4c271c1c

__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=JJfBgwamb9qYHyOBZm3gWmkEPYaesJx
%2F6AknH8y6Jjh3oAemh1tJOs40vwCH4zBewhIPBtK1%2B8YtEiJfmQ9lywmrwccRwzRNONm
%2Bh44T4DNp6bMG5esXtxqm18b5tvrDRhbWFP9jUTmXZE057Qs9xtx7QN3bP00jHm8i4ytwJvowMsv89H4S5ZNw%2BsGZyhA50pYCFH9tc%2BnMT9DrlrAhWQ
%3D%3D&__VIEWSTATEGENERATOR=6603A2E4&__EVENTVALIDATION=H6sgsm3N2nI9J
%2BVip5NrIjZtB7cnjD42oMQ3hE7TAInxrRS4WPkcjPmE5zqgN8GTM2zncxechZMlDDFA%2FlcjW%2F%2FrVayNpaI3nxtBSmc
%2FNNZoCxbmYF2GrOjSDEY2YNFvvk
%2BTu7yZ3kb5gwkTpiyOylOl3uUFO3%2Fg8UgJE2NFVXCiCyXSp9LqaQmlXmvLXH3%2FaV98f3FjUc7rc6WfLrxtAXLvj5z0qcbml9HtmjSXo8qgDYXW3VVWMG3
xmdNjk6Ms&ctl00%24MainContent%24TxtID=400301&ctl00%24MainContent%24TxtName=nDKoIVJIL4YIJoo7jd%2FqdPyNWHxCzYosLYYWd2aGX9g
%3D&ctl00%24MainContent%24TxtAge=hP9aDI9u9M%2FCgya4WD%2FpqQ%3D%3D&ctl00%24MainContent%24TxtAddress=XerqgDPcEhptb
%2FWgsMK1FjSPepdxa%2BnE2hYajYM%2B4Oc%3D&ctl00%24MainContent%24Button1=Save+DataHTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=utf-8
Content-Encoding: gzip
Vary: Accept-Encoding
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Mon, 28 Mar 2016 20:41:21 GMT
Content-Length: 2654

...........Yys.J..{S...h........-..
....R.....B.....w...1..N..Z....==.X6....].C..5_.YL..........R...7...K... ..*....q...
..........x.-...h...h..&A..._.A6.|.45...g]..A......A.....e.Q.1.q.c..c..&.P.L$
```

Packet 3. 2 client pkt(s), 2 server pkt(s), 1 turn. Click to select.
Entire conversation (4441 bytes)
Show data as ASCII
Stream 0
Find:
Find Next

**Figure 7 Encrypted data of the client with ID: 700701**

The two scenario A and B sent data over a transmission medium to a dedicated server. In scenario A, we were able to transmit data without encrypting it, hence the data stored in the server was in plaintext. Whiles in scenario B, an AES 256bits key length was used to encrypt the data before transmitting to the server. The aim of this research was achieved in in scenario B, thus encrypting data with AES 256bits key length before it is transmitted to the server. An organization that do not deploy encryption in its daily business transactions or data transfer stand a higher risk of losing very critical and sensitive information hackers as seen in scenario A. The results from the unencrypted data were obvious, as all the input data were transmitted in it plaintext format to the server. This possess a great danger to the organization, especially about the data of its customers and partners, e-commerce and other business transactions.

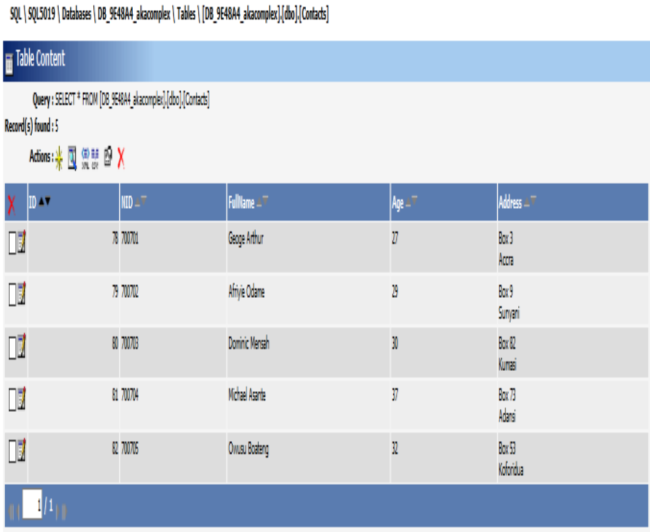
Regarding scenario B, the input data was scrambled by the AES 256bits key length algorithm before transmission. When malicious people intercept the data, because it has been scrambled by the AES 256-bit algorithm, they cannot read the ciphertext unless they crack it first. For a hacker to be able to crack the key of the encrypted data, it will require of him enormous tasks that span across several period of time, thus, 256 bits AES application: 2256 = 1.1x 1077 possible keys. It is assumed an attacker with the capability to build or purchase a system that tries keys at the rate of one billion keys per second requires many centuries to crack AES. With this hypothesis, the attacker would require about 1x1021years to try all likely keys for the weakest edition of AES-256.

Since the application uses AES encryption and decryption key size 256 bits and a cipher block size 256 bits, the ciphertext is so strong that it becomes virtually impossible for any malicious person to decrypt it. Since the ciphertext was encrypted with AES 256-bit and key size of 256 bits, a bad person will requires 2256 possible keys to be able to use brute force to decrypt a character of the key.

### 4.1 Results for unencrypted data in the server

Table 2 shows the stationary data saved in the server which clearly indicates the system did not implement any form of encryption which means all the data saved are in plaintext under section (A) category. It is very risky and extremely dangerous to transmit sensitive information without proper measure to protect or shield it. This implies that any hacker getting access to the save can easily get hold of the organization data in its raw (plaintext) format.

**Table 2Unencrypted data saved in the server**



SQL \ SQL5019 \ Databases \ DB_9E48A4_akacomplex \ Tables \ [DB_9E48A4_akacomplex].[dbo].[Contacts]

Query : SELECT * FROM [DB_9E48A4_akacomplex].[dbo].[Contacts]

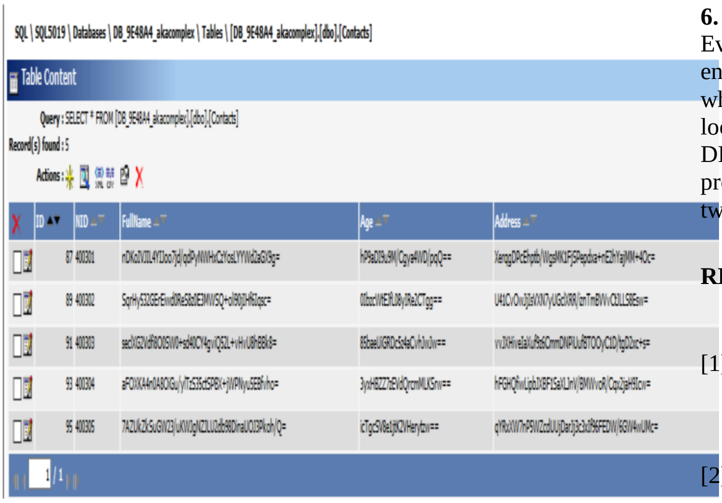
Record(s) found : 5

| ☒ | ID ▲▼ | NID | FullName | Age | Address |
|---|---|---|---|---|---|
| □ | 78 | 700701 | Geoge Arthur | 27 | Box 3<br>Accra |
| □ | 79 | 700702 | Afriyie Odame | 29 | Box 9<br>Sunyani |
| □ | 80 | 700703 | Dominic Mensah | 30 | Box 82<br>Kumasi |
| □ | 81 | 700704 | Michael Asante | 37 | Box 73<br>Adansi |
| □ | 82 | 700705 | Owusu Boateng | 32 | Box 53<br>Koforidua |

### 4.2 Results for encrypted data in the server

Table 3 shows the saved data of an organization that transferred data in an encrypted. The input data, the intercepted data and the data saved in the server are different. The input data has been scuttled by the AES 256 bits block cipher algorithm hence, the saved data in the server has been completely protected by the application.

**Table 3 Encrypted data saved in the server**



SQL \ SQL5019 \ Databases \ DB_9E48A4_akacomplex \ Tables \ [DB_9E48A4_akacomplex].[dbo].[Contacts]

Table Content

Query : SELECT * FROM [DB_9E48A4_akacomplex].[dbo].[Contacts]

Record(s) found : 5

| ID | NID | FullName | Age | Address |
|---|---|---|---|---|
| 87 | 400301 | nDKoIVIIL4YIJoo7jd/qdPyNWHxCzYosLYYWd2aGX9g= | hP9aDI9u9M/Cgya4WD/pqQ== | XenggDPcEhptb/WgsMK1FjSPepdxa+nE2hYajMM+4Oc= |
| 89 | 400302 | SqrHy532GErEiwd0ReS8dE3MWSQ+dl90jJHf6Iqsc= | 0IbzcWtEJfL08yIRe2CTgg== | U41CvOwJjIsVXN7yUGdXRR/znTmBVVvCt1LLS8Esw= |
| 91 | 400303 | secIXG2Vdfi8OXSW0+sd40CY4gviQ52L+vHvU8hBBk8= | 85baeUGRDc5s4aCvhJwJw== | vvJXHiveIaXuf9t6CmmDNPlUuf8TOOyC1D/tgD2xc+s= |
| 93 | 400304 | aFOXXA4n0A8OIGu/yITzS35ctSPBX+jWPNyu5EBfvho= | 3yxHBZZ7tEVdQrcmMLKSrw== | hFGHQfiwLipbJXBF1SaXLJnV/BMWvoR/Cqx2jaH9Icw= |
| 95 | 400305 | 7AZUkZkSuGW23/uKWJgNZ1LU2db98DinaUOJ3Pkoh/Q= | icTgcSV8e1jtK2VHerytzw== | qYRxXW7nP5WZcdUUjDarJj3c3xI96FEDW/6GW4wUMc= |

## 5. CONCLUSION AND RECOMMENDATION

The conclusions of this research could be summarized as follows:

- ✓ The tests conducted indicate a high level security for data in transit and stationary. The data protection using AES is to provide optimized data security to classified and non-classified data. The test conducted for all the five client (5) using the developed web application indicated a successful data protection. When encryption algorithm was use to encrypt the data, it was realized that data in transit, data saved in the server are highly protected and major data losses.
- ✓ The results also results of the data seen in the server shows that even if a hacker hacks or intercepts the data through a hacking tool or social engineering, the data will be meaningless the him/her. This research will solve problem of organizations losing their sensitive data to unauthorized persons, a 256 bits key length offers more security to data both at rest and in transit [3].
- ✓ Organizations with virtual offices at remote location require this form of application to enable their remote workstations to communicate securely with the server. Since the ciphertext was encrypted with AES 256-bit and key size of 256 bits offers a better and a more secured that as compared with [19] [15] [16] [16] [3] that employed 128 key length, anyone who want to crack an AES with 256 key length will requires $2^{256}$ possible keys to be able to use brute force to decrypt a character of the key. An average throughput 1054Mbit/sec for encryption and 615 Mbit/sec for decryption was realized from the test analysis.

### Recommendation

Due to the high level security capabilities provided by AES with 256 bits key length, we recommended to organization such as school, banks, microfinance and churches, that seeks to do a secure business transaction both online or on corporate network infrastructure to protect the data of its clients, staff, partners and supplier with AES with 256 bits key length algorithm.

## 6. FEATURE WORKS

Even though the AES algorithm offers data security and better encryption as shown by this paper, we wish that researchers who wish to extended and improve upon this work should look at combining two encryption algorithm such as AES and DES to give much and better organization data security and protection from unauthorized persons and hackers, since these two methods have unique features.

## REFERENCES

[1] Ernst & Young, "Data loss prevention: Keeping your sensitive data out of the public domain," Ernst & Young Global Limited, UK, 2011.

[2] A. Ibrahim, "FPGA-based Hardware Implementation of Compact AES Encryption Hardware Core," *WSEAS transactions on circuits and systems. ISSN: 2224-266X*, vol. 14, 2015.

[3] B. ,. Madhuri and N. M. Suresh, "Implementation of Advanced Encryption Standard Algorithm for Communication Security Using FPGA," *International Research Journal of Engineering and Technology (IRJET)*, vol. 03, no. 07, pp. 1176-1179, 2016.

[4] F. Twum, k. Nti and M. Asante, "Improving Security Levels In Automatic Teller Machines (ATM) Using Multifactor Authentication," *International Journal of Science and Engineering Applications*, vol. 5, no. 3, pp. 126-134, 2016.

[5] G. C. Kessler, "An Overview of Cryptography," 2008. [Online]. Available: http://www.garykessler.net/library/crypto.html. [Accessed 27 January 2015].

[6] V. Beal, "cryptography," 2014. [Online]. Available: http://www.webopedia.com/TERM/C/cryptography.html. [Accessed 14 June 2015].

[7] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson, T. Kohno and M. Stay, "The Twofish Team's Final Comments on AES Selection," 15 May 2000. [Online]. Available: https://www.schneier.com/academic/paperfiles/paper-twofish-final.pdf. [Accessed 2 Febuary 2016].

[8] W. M. Tatun, "The Advanced Encryption System (AES) Development Effort: Overview and Update," SANS Institute, 2001.

[9] M. Pitchaiah, D. Philemon and Praveen, "Implementation of Advanced Encryption Standard Algorithm," *International Journal of Scientific & Engineering Research*, vol. 3, no. 3, pp. 1-6, 2012.

[10] J. Nechvatal, E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti and E. Roback, “Report on the Development of the Advanced Encryption Standard (AES),” Computer Security Division Information Technology Laboratory National Institute of Standards and Technology Technology Administration U.S. Department of Commerce, U.S.A, 2000.

[11] T. Alaa, A. Zaidan and B. Zaidan, “New Framework for High Secure Data Hidden in the MPEG Using AES Encryption Algorithm,” *International Journal of Computer and Electrical Engineering,* vol. 1, no. 5, pp. 1793-8163, 2009.

[12] Wikipedia, “Rijndael S-box,” 11 November 2013. [Online]. Available: https://en.wikipedia.org/wiki/Rijndael_S-box#cite_note-1. [Accessed 12 January 15].

[13] G. Mohan and K. Rambabu, “An Efficient FPGA Implementation of the Advanced Encryption Standard Algorithm,” *International Journal for Scientific Research & Development (IJSRD),* vol. 2, no. 07, pp. 413-417, 2014.

[14] C. Paar and J. Pelzl, “Advanced Encryption Standard,” 2009. [Online]. Available: https://en.wikipedia.org/wiki/Advanced_Encryption_Standard . [Accessed 25 May 2016].

[15] T. Hoang and V. L. Nguyen, “An efficient FPGA implementation of the Advanced Encryption Standard algorithm,” *In Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), IEEE RIVF International Conference,* pp. 1-4, 2012.

[16] S. ,. R. Kumar and V. Viswanadha, “An efficient FPGA implementation of the AES Algorithm With Reduced Latency,” *International Journal for Scientific Research & Development (IJSRD ),* vol. 1, no. 10, pp. 2074-2077, 2013.

[17] K. Gayathri and W. Yasmeen, “Data Encryption and Decryption using AES with Key Length of 256 Bits,” *International Journal of Scientific Engineering and Technology Research,* vol. 03, no. 20, pp. 4143-4146 , 2014.

[18] C. Sadashiva and S. Sunkari, “Data Encryption and Transition by AES Algorithm with UART,” *International Journal of Scientific Engineering and Technology Research,* vol. 03, no. 35, pp. 6935-6938, 2014.

[19] P. Aatheeswaran and R. Babu, “FPGA can Be Implemented by Using Advanced Encryption Standard Algorithm,” *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering,* vol. 2, no. 1, pp. 675-679, 2013.

[20] P. Anitha and Palanisamy, “Data Protection Algorithm Using AES,” *International Journal of Current Research,* vol. 33, no. 6, pp. 291-294, 2011.

[21] N. Ahmad, R. Hasan and W. Jubadi, “Design of AES S-Box using combinational logic optimization,” *IEEE Symposium on Industrial Electronics & Applications (ISIEA),* pp. 696-699, 2010.