International Journal of Advancements in Technology

Nti et al., Int J Adv Technol 2017, 8:2
DOI: 10.4172/0976-4860.1000183

Research Article Open Access

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

Isaac Kofi Nti[*], **Eric Gymfi and Owusu Nyarko**

Department of Electrical/Electronic Engineering, Sunyani Technical University, Ghana

[*]**Corresponding author:** Isaac Kofi Nti, Department of Electrical/Electronic Engineering, Sunyani Technical University, Ghana, Tel: 0208736247; E-mail: nti.k.i@stu.edu.gh

Received date: February 23, 2017; **Accepted date:** March 09, 2017; **Publication date:** March 16, 2017

Copyright: © 2017 Nti IK, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License; which permits unrestricted use; distribution; and reproduction in any medium; provided the original author and source are credited.

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. 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

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 [1-3].

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.

Cryptography

The science of writing or transforming the meaning of data into secret data to prevent people from reading is generally termed as Cryptography [5]. The history of the earliest use of cryptography was documented close to 1900 B.C., associate Egyptian academician was ready to generate and much used a non-standard symbols in writing messages. Cryptography became thus common throughout the missives wartime era wherever battle plans and communication ways were predominate within the earlier period. Once the utilization of laptop and alternative device were introduced into the communication system, there was the necessity for a correct management of data keep on these devices. In information, communications and networking, cryptography is important once human action over a transmission medium, particularly the un-trusted medium, significantly the net [5].

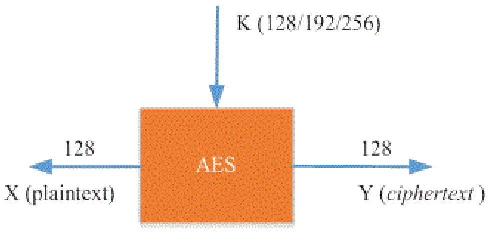


Figure 1: Cryptography process.

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.

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

Description of AES algorithm

On January two, 1997, government agency National Institute of Standards and Technology (NIST) declared the initiation of an attempt to develop the AES and created a proper concern algorithms on September twelve, 1997 [7]. 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. 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 internal structure

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.

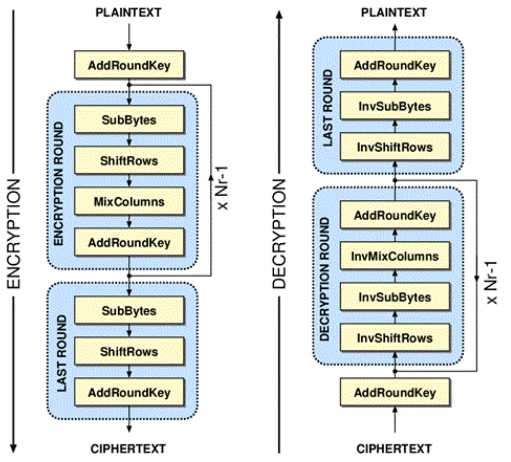


Figure 2: Overall structure of AES algorithm (Source: http://crypto.stackexchange.com/questions/8043/aes-addroundkey).

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 [7,10]. 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 and addRoundkey.

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.

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.

		x															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
	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
y	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

Table 1: S-box table.

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. For additional knowledge in S-box table calculations refers to [11]. S-box table as shown in Table 1.

Shift Rows: In Shift Rows 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 [12].

Mix Column: From the Mix Column operations, there is a transposition of linear transformation made to join the 4-byte in each column as shown in Figure 2. 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 Figure 2. Addition is simply XOR. Multiplication is modulo irreducible polynomial [13,14]. 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,

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

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

$$(0 \le c \le N_b)$$

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 (Figure 3). The subkey is inserted by combining every byte in the state with it related byte in the subkey by means of bitwise XOR [15].

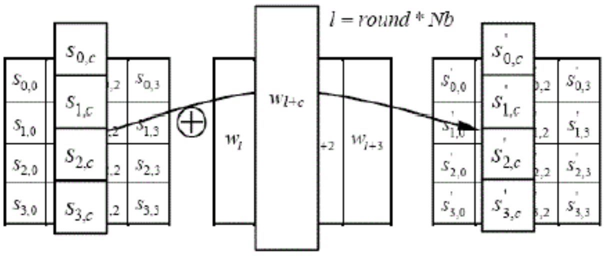


Figure 3: addRoundKey XORs each column of the State with a word from the key schedule.

Related works

A FPGA-based implementation of the Advanced Encryption Standard (AES) algorithm was proposed by [15]. 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 1054 Mbit/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 [16] to advance the security of data in motion. [17] 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 [14]. 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 [14]. An implementation of AES encryption and decryption standard AES-128 was proposed by [18]. 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 [18]. Proposed implementation of the 128 bit AES normal on a Field Programmable Gate Array (FPGA) 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], hence this paper focuses on the implementation of the AES algorithm with a 256 bits key length to prevent data loss in an organization.

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.

Design concept: Figure 4, shows the pictorial view of the proposed layout for the implementation of the AES algorithm with 256 bits key length for organizational data protection.

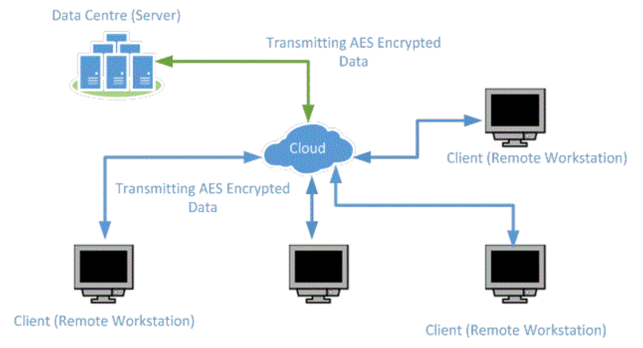
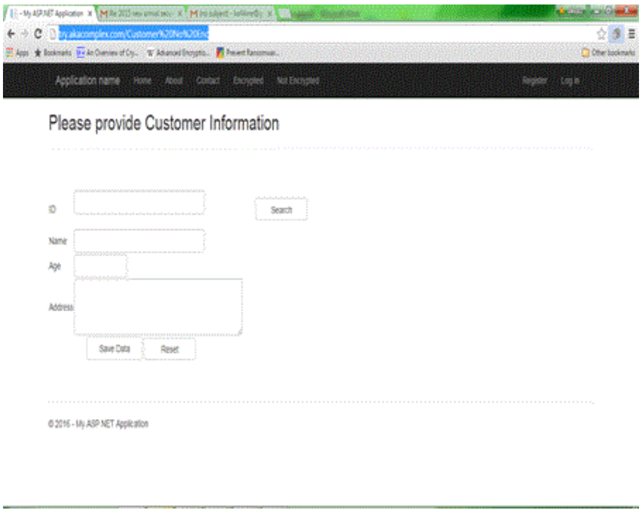

Data Centre (Server)
Transmitting AES Encrypted Data
Cloud
Client (Remote Workstation)
Transmitting AES Encrypted Data
Client (Remote Workstation)
Client (Remote Workstation)

Figure 4: Conceptual design of proposed system.

A web application with login details of client as shown in Figure 5 was developed to collect raw data in plaintext.



Please provide Customer Information

ID
Search
Name
Age
Address
Save Data
Reset

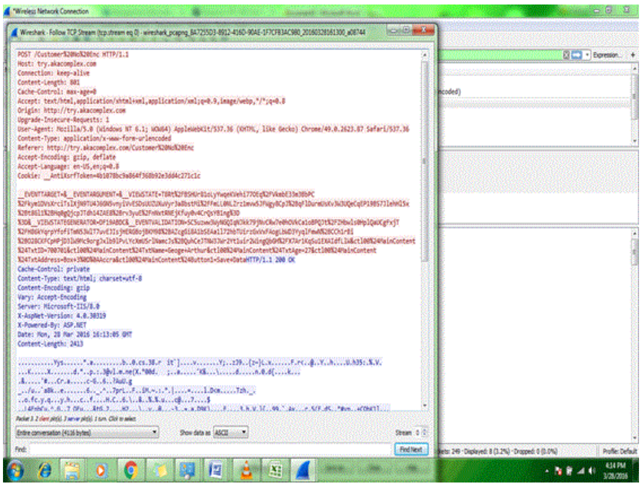
© 2016 - My ASP.NET Application

Figure 5: Plaintext data collection interface.

Results and Discussion

Five different data was collect from the front end for testing purpose. The data transmission from the client machine to server was done in two section A and B. In section 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.

Same client with ID 700701 data was transmitting under section 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-90AE-1F7CF83AC980_20160328161300_a08744

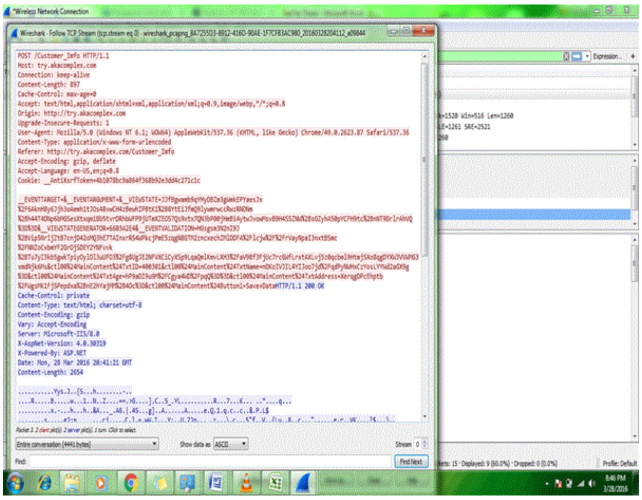
```
POST /Customer%20No%20Enc HTTP/1.1
Host: try.akacomplex.com
Connection: keep-alive
Content-Length: 801
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Origin: http://try.akacomplex.com
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (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=[illegible]
[illegible]
[illegible]
%3D&__VIEWSTATEGENERATOR=DF19A0DC&__EVENTVALIDATION=[illegible]
[illegible]
[illegible]&ctl00%24MainContent
%24TxtID=700701&ctl00%24MainContent%24TxtName=Gegoe+Arthur&ctl00%24MainContent%24TxtAge=27&ctl00%24MainContent
%24TxtAddress=Box+3%2BD%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;..z39..[z=]L.x.......F.r<..@..Y..h....U.h35:.%.V.
...K......X.......d.*..p..:.3@vl.e.ne(X.*00d.   ;..a......'K&...\......d.....n.0.d(....k...
.&.....'#...Cr.a.....c-6..6..?AuO.g
_../u..`a8k..e.......6.._^..7prL..F..iM.~.:.*.|....•....1.Dcm.....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 (4126 bytes) ▾ Show data as ASCII ▾ Stream 0
Find: Find Next

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


*Wireless Network Connection
Wireshark · Follow TCP Stream (tcp.stream eq 0) · wireshark_pcapng_8A7255D3-8912-416D-90AE-1F7CF83AC980_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=[illegible]
[illegible]
[illegible]
%3D%3D&__VIEWSTATEGENERATOR=6603A2E4&__EVENTVALIDATION=[illegible]
[illegible]
[illegible]
[illegible]
[illegible]&ctl00%24MainContent%24TxtID=[illegible]&ctl00%24MainContent%24TxtName=[illegible]
%3D&ctl00%24MainContent%24TxtAge=[illegible]&ctl00%24MainContent%24TxtAddress=[illegible]
[illegible]&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....==.>G....].C..S_YL..........R...7...K... ..*....q...
..........x.-....h...h..&A..._A6.[.45...g]..A......A.....e.Q.1.q.c..c..&.P.L$
```

Packet 3. 1 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

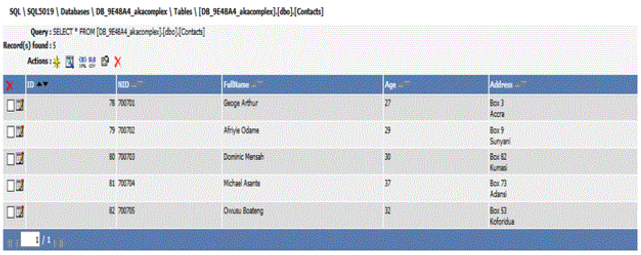
Figure 7: Encrypted data of the client with ID: 700701.

Results for unencrypted data in the server

Figure 8 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.

Results for encrypted data in the server

Figure 9 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.



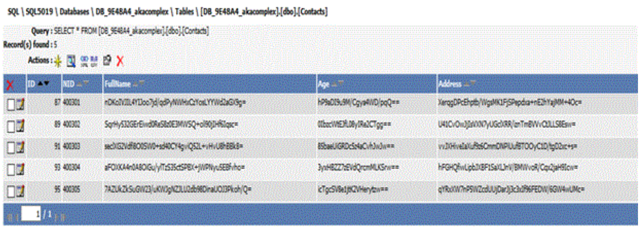
SQL \ SQL5019 \ Databases \ DB_9E48A4_akacomplex \ Tables \ [DB_9E48A4_akacomplex].[dbo].[Contacts]

Query : SELECT * FROM [DB_9E48A4_akacomplex].[dbo].[Contacts]
Record(s) found : 5
Actions :

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

1 / 1

Figure 8: Unencrypted 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
Actions :

☒	ID	NID	FullName	Age	Address
□	87	400301	nDKoIVIIL4YIJoo7jd/qdPyWWHsCzYosLYYWd2aGX9g=	hP9aDI9u9M/Cgya4WD/pqQ==	XerqgDPcEhptb/WgsMK1FjSPepdxa+nE2hYajMM+4Oc=
□	89	400302	SqrHy532GErEwd0Ra5B20E3MWSQ+ol90jIHfiIqsc=	0IbzcWtEJfL08yIRa2CTgg==	U41CvOwJjJdWXN7yUGdXRR/anTmBWvQ1LLS8Esw=
□	91	400303	secIXG2VdfBO0SW0+sd40CY4gvQ52L+vHvU8hBBk8=	8SbaeUGRDcS24aCvhJwJw==	vvJXHive1aXuf9s6CmmDNPIUuf8TOOyC1D/tgD2sc+s=
□	93	400304	aFOXKA4n0A8OIGu/yITzS35ctSPBX+jWPNyu5EBfvho=	3yxHBZZ7tEVdQrcmMLKSrw==	HFGHQfwLipbJXBF1SaXLJnV/BMWvoR/Cqx2jaH9Icw=
□	95	400305	7AZUkZkSuGW23/uKW3gNZ3LU2db98DinaUOJ3Pkoh/Q=	kTgc5V8e1jtK2VHerytzw==	qYRxXW7nP5WZcdUUjDarJj3c3xJf96FEDW/6GW4wUMc=

1 / 1

Figure 9: Encrypted data saved in the server.

Conclusion

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 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 [3,14,15,18,19] 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 1054 Mbit/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.

References

1. Ernst, Young (2011) Data loss prevention: Keeping your sensitive data out of the public domain. Ernst & Young Global Limited, United Kingdom.
2. Ibrahim (2015) FPGA-based hardware implementation of compact aes encryption hardware core. WSEAS transactions on circuits and systems.
3. Madhuri A, Suresh NM (2016) Implementation of advanced encryption standard algorithm for communication security using FPGA. Int Res J Engg Tech 3: 1176-1179.
4. Twum F, Nti K, Asante M (2016) Improving security levels in Automatic Teller Machines (ATM) using multifactor authentication. Int J Sci Engg Appl 5: 126-134.
5. Kessler GC (2008) An overview of cryptography.
6. Beal V (2014) Cryptography.
7. Nechvatal J, Barker E, Bassham L, Burr W, Dworkin M, et al. (2000) Report on the development of the Advanced Encryption Standard (AES). Computer security division information technology laboratory national institute of standards and technology administration, U.S. Department of Commerce, USA.
8. Tatun WM (2001) The Advanced Encryption System (AES) development effort: Overview and update. SANS Institute.
9. Pitchaiah M, Philemon D, Praveen (2012) Implementation of advanced encryption standard algorithm. Int J Sci & Engg Res 3: 1-6.
10. Alaa T, Zaidan A, Zaidan B (2009) New framework for high secure data hidden in the MPEG using AES encryption algorithm. Int J Comp Electrical Engg 1: 1793-8163.
11. Ahmad N, Hasan R, Jubadi W (2010) Design of AES S-Box using combinational logic optimization. IEEE Symposium on Industrial Electronics & Applications , pp: 696-699.
12. Mohan G, Rambabu K (2014) An efficient FPGA implementation of the advanced encryption standard algorithm. Int J Sci Res Development (IJSRD) 2: 413-417.
13. Paar C, Pelzl J (2009) Advanced Encryption Standard.
14. Hoang T, Nguyen VL (2012) An efficient FPGA implementation of the Advanced Encryption Standard algorithm, 2012 IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), IEEE.
15. Kumar SR, Viswanadha V (2013) An efficient FPGA implementation of the AES algorithm With reduced latency. Int J Sci Res Development 1: 2074-2077.
16. Gayathri K, Yasmeen W (2014) Data encryption and decryption using AES with key length of 256 bits. Int J Sci Engg Tech Res 3: 4143-4146.
17. Sadashiva C, Sunkari S (2014) Data Encryption and Transition by AES Algorithm with UART. Int J Sci Engg and Tech Res 3: 6935-6938.
18. Aatheeswaran P, Babu R (2013) FPGA can Be Implemented by Using Advanced Encryption Standard Algorithm. Int J Advanced Res Elec Electr Inst Engg 2: 675-679.
19. Anitha P, Palanisamy V (2011) Data Protection Algorithm Using AES. Int J Current Res 33: 291-294.