Rijndael Encryption Algorithm A Solution To Private Issues Of Internet of Things

Ayannusi A.O[1] , Oloyede E.O[2] Olusanya O.J[3]
[1](Cisco Department, Ogun State Institute of Technology Igbesa, Nigeria)
[2](Cisco Department, Ogun State Institute of Technology Igbesa,, Nigeria)
[3](Cisco Department, Ogun State Institute of Technology Igbesa,, Nigeria)

Abstract— *Because of the high complexities issues in the Internet of Things and its applications, there is a need to propose an appropriate security model that would help in managing and controlling it. This paper critically examined Rijndael's encryption algorithm in terms of design choices, method of implementation, performance, strength, weaknesses and to determine its applicability in the Internet of things devices and applications.*

In this paper, Round transformation will be studied extensively so as to and know and determine the number of rounds that would be added to strengthen the encryption method. With the sole purpose of being able to make a recommendation on its use in the context of IoT with their requirement.

Index Terms— Encryption, Cryptography, IoT, Rijndael, Round transformation

1 INTRODUCTION

Due to the vulnerability of DES to brute force attacks, it was necessary to replace DES. In January 1997 when NIST National Institute of Standard Technology revealed that it was looking for an Advanced Encryption Standard that will replace DES. A call was officially made for algorithms in which fifteen candidates were considered. These were later screened to five finalists, MARS Rivest's RC6, Rijndael, SERPENT, and TWO FISH [1].

Since Rijndael's adoption in 2001 as the national standard for encrypting sensitive data, the Advanced Encryption Standard became one of the most popular and widely used Encryption standard all over the world. Though there was no exploitable weakness in the AES, more recently some cryptographers have begun to theorize the existence of an algebraic weakness in the cipher. [2]. And again, Rijndael selection was subjected to criticism that the algorithm is not as secure compared to some other choices. Though, its criticism is correct theoretically, but cannot be verified that using this algorithm to secure data will be susceptible to attack.

When AES was selected many years ago, the digital technologies were quite different from now and the magnitude of the challenges was less, so with recent advanced technology and the emergence of new applications like Big data's application, Internet of Things in addition to other applications, it has become a necessity for designing a new contemporary algorithm for the current demands. However, no one denies that the selection of Rijndael at that time was a good choice for civil application on software and hardware implementation and on many various platforms but the excessive speed for information technology progress leads to take into account recalculation of the security level for the current and prospective future requirement. [3].

The AES was to be a block cipher symmetric technique that will support keys sizes of 128, 192, and 256-bit keys. Rijndael is built on substitution and permutation networks. It does not use the Feistel network. It is more secure than DES and hard to crack. AES is more complex than DES but it is fast and very efficient. It works with 128- bit fix block size plain text and variant key sizes [4].

The Rijndael algorithm is a symmetric block cipher that shows a higher level of modular design, that would allow any changes to resists any future attack in an easy way, unlike the old algorithm designs. The choice of AES was going to be a problem balancing several factors like overall efficiency, security, and performance. As such, it was not likely that the choice of any other algorithm would be fully accepted from all quarters.

Rijndael can be executed effectively in software on various types of processors and utilize a limited set of instructions and has little adequate parallelism to fully exploit new pipeline multi ALU processors because of its suitability for most applications [5]. low RAM, a small number of cycles, speed, and its execution on various types of processors and dedicated hardware would make it suitable for small embedded IoT devices. The algorithm is written in such a way that block length and/or key length can easily be extended in multiples of 32 bits and it is specifically designed for efficient implementation in hardware or software on a range of processors. The design of Rijndael was greatly affected by the cipher block called Square [1].

The IoT is comprised of smart machines, interacting and communicating with other machines, objects, environments, and infrastructures. As a result, huge volumes of data are being generated and data is being processed into useful actions that can "command and control" things to make our lives much easier and safer and also to reduce our impact on the environment [6]. However, [7], one of the biggest stumbling blocks to IoT development is likely to be concerned about privacy and security. It has been found by Hewlett-Packard Packard (HP) that 70% of the most commonly used IoT device contains security vulnerabilities.

In addition, the IoT also poses significant security challenges that need to be addressed; a simple firewall is no longer sufficient and can be attacked via wireless channel directly [8]. The possibilities of the Internet of Things seem to be endless when we zero in on some of the capable gadgets that can be part of the pool of existing things that are connectable.

I. Related work

1. The Design rationale of Rijndael

Resistance against all known attacks, code compactness, speed, all familiar attacks, and Design clarity on a wide range of platforms are three keys design simplicity of Rijndael. It is an iterated block cipher with a variable block length and a variable key length. The block length and the key length can be separately namely 128, 192, or 256 bits. [1]. Rijndael design is based on security and efficiency through simplicity and symmetry. This simplicity in Rijndael was achieved through symmetry, parallelism, modularity, and lack of arithmetic operation which not only affects security but efficiency as well [9].

As a result of design simplicity, resistance against all known attacks, and its use on a broad range of platforms; Rijndael can go with a very light key schedule that requires little memory which can also bring about the implementation of secure and efficient implementation of the cipher on smart card and low memory devices [10].

[11] the criteria of security and efficiency are applied by all cipher designers and there are situations in which efficiency is sacrificed to obtain a higher security margin. The challenge is to bring up a cipher design that offers a reasonable security margin while optimizing efficiency. Rijndael and its related ciphers are instances of a ciphers family that allow large flexibility in block length without losing the properties of and high resistance against cryptanalysis.

1.1 Method of Implementation

Since the adoption of Rijndael as Advanced Encryption Standard, many programming languages have been used in implementing Rijndael. The two fastest software implementations available for a PC platform are B.Gladman and H. Lipman and thus fixes both block length and the key length of Rijndael 128 bits [10]. Also according to [9]. Implementation are available in many different programming languages. As AES standard is open, organizations or users who wish the Rijndael algorithm are free to do so. AES standard can be programmed in software or built with pure hardware.

[12], implements Rijndael block ciphers in reconfigurable hardware based on simple digital design rules applied to the iterated block cipher. Multiplexor model, RAM-based implementation and composite field solution are different ways proposed by [12], to implement Rijndael algorithm.

However, Field Programmable Gate Arrays (FPGA) offer a quicker more customizable solution. It can be implemented with reasonable efficiency on an FPGA, with the encryption and decryption taking an average of 320 and 340 as respectively for every 128 bits [13].

Rijndael appears to continually perform well in software and hardware over a broad range of computing situations regardless of its use in feedback and non-feedback mode.[9] Efficient implementation of Rijndael has been realized in software and dedicated hardware. Research and experiments have been done to make this implementation secure against side-channel attacks [11]. Likewise, [14], Efficient implementation of Rijndael has been realized in software and dedicated software.

[15], Rijndael can be executed very effectively on broad types of processors for instance smart cards. It has the quickest encryption and decryption time and gives the greatest result when both hardware and software implementation were taken into consideration.

Rijndael encryption technique is a block symmetry cipher that uses a symmetric key and can be 128,192 or 256 data blocks of 128 bits encryption. Rijndael encrypts the data block in 10, 12, and 14 rounds depending on the key size. It is fast and flexible. It is not only faster executed in both hardware and software, but it is also the latest algorithm that is required by the United State and international standards and also more secure to use. And again, it supports larger key sizes than other algorithms [16].

1.1 Hardware Implementation

Both academia and business concentrated on the design of Rijndael cores in reconfigurable hardware. This implementation indicates that fellow reconfigurable platforms with their great distributed memory structure are well suitable for Rijndael prototype operations and most of them used precomputed subkeys. It can be implemented very efficiently on a wide range of processors and in hardware [14]. Many hardware of Rijndael encryption algorithm using VHDL is available [17]. Rijndael can also be implemented on an 8-bit microcontroller [14].

Rijndael very low memory requirement makes it adequately suitable for restricted space conditions, in which it also demonstrates excellent performance [9] In dedicated hardware, Xtime can be implemented with the combination of a hardwired bit transportation and four XoR gates. The SubBytes step is the most critical part of the hardware implementation. However, when building dedicated hardware for supporting both encryption and decryption, the required chip area is relevant and can also be limited by using parts of the circuit for both transformations [10].

1.2 Software Implementation

Software implementation has been generated with manufactured assembly for the greatest accomplishment. Software execution needs 232 cycles for 10 rounds of 128 data bit bocks likewise 128-bit keys on a pentiumII and 124 cycles on 1A64. And because all these operate on processors that are in the GHz range, this performance subsequently approaches the effectiveness of a special hardware implementation by folllowing per under the power utilization. Though, the software remedy on normal purpose processors is still at least two orders of magnitude higher than special hardware implementation.

In a software implementation, the encryption and decryption speed and the required amount of working memory and program storage memory are relevant [11]. Software has been designed which advertise clock speed ranging from 200 to 350MHz (2002)

1.3 Performance of Rijndael

One of the key reasons for Rijndael to be selected was its good performance across different platforms from 8bit smart cards to standard 32-bit computer processors as well as having the potential for fast dedicated hardware [9]. The performance of Rijndael is inversely proportional to key size and this key size will increase the performance. [18].

The two major operations that affect the performance of Rijndael are key setup and the operation that uses the keys to encrypt the data. In order words, performance is a function of key setup and the manipulation of data that is done in the rounds of an algorithm. As a result, the key agility and the throughput based on rounds affect the overall performance [19].

When Rijndael is compared with other security algorithms DES, 3DES, and RC2 it simply shows that Rijndael outperformed other algorithms in both the number of requests, processes per second in different user loads and in response time, in different user-load situations [20]. At the same time, Rijndael has an advantage over Blowfish and RC6 in terms of memory usage, performance, and throughput [21].

There is every possibility and potential to speeding up Rijndael depending on how it is implemented in the various languages [9].

2.1 Implementation aspects

- Rijndael can be executed to operates at speeds uncommonly fast for a cipher block on a Pentium (Pro). There is a trade-off between performance and table size.
- Rijndael can also be executed on a Smart Card in a small amount of code, using a small amount of RAM and taking a little number of cycles. There is some ROM/performance trade-off.
- The round modification is parallel by design, a significant benefit in future processors and specialized hardware.
- Because cipher does not utilize arithmetic operations, it has no unfairness towards large or small endian processor structures.

The Simplicity of Design:

- The block cipher is fully self-supporting. It avoids the utilization of other cryptographic elements, S-boxes “lent” from fine-recognized ciphers, bits acquired from Rand tables, digits of π.
- The block cipher avoids hinging its security or part of it on the vague and not incomprehensible relationship between arithmetic workings.

- The tight cipher blueprint avoids suf-

ficient room to hide a trapdoor. Variable block

length:

- The block lengths of 192 and 256 bits permit the building of a collision-resistant iterated hash function using Rijndael as the compression function. The block length of 128 bits is not deemed adequate for this aim presently.

Extensions:

- The design permits the provisions of different forms with the block length and key length both ranging between 128 to 256 bits in steps of 32 bits.
- Albeit, the number of rounds of Rijndael is defined in the specification, it can be altered as a variable in event that there are security issues. A secured encryption or decryption system key is constantly obtained from the cipher key [22].

1.2 Weakness of Rijndael

When the AES was selected few years ago, the digital technologies were quite different from now and the magnitude of the challenges was less, so with recent advanced technology and the emergence of new applications like big data applications and IoT applications [3]. The selection of Rijndael as the AES at that time was no doubt a good choice for public application on software and hardware implementation and many of various platforms but the excessive rate at which Information Technology progress leads to take into account recalculation of the security level for the current and prospective future requirement [3].

Using a method termed a biclique attack, they can reclaim AES keys quicker than a brute-force attack by a factor of between three and five, depending on the cipher version. However, even this attack does not intimidate the efficient utilization of AES due to its high computational complexity.

AES has demonstrated to be a dependable cipher, and the only real outstanding attacks against AES have capitalized on side-channel attacks on the weaknesses detected in the execution or key management of particular AES-based encryption products [23]. [24] Rijndael has been developed to have a powerful protection against classical attacks, such as linear cryptanalysis, differential cryptanalysis, etc. Even though, Rijndael is developed from the square algorithm and is very algebraic, new algebraic and enhanced differential attacks have surfaced. The principal weakness of Rijndael is the issue of linearity in the key schedule and S-box. Apart from the fact that it is difficult to the details of the process of Rijndael because it is too patents encryption, it will also be difficult to decrypt data if the secret key is lost [16].

1.3. Round Transformation

Rijndael is an iterated cipher block, the encryption or decryption of a block of data is achieved by the iteration of a peculiar transformation. With a repeated block cipher, the different transformations work in sequence on intermediate cipher results i.e. states [19].

[10], The round transformations of Rijndael and its steps operate on an immediate result, called states. The state is a rectangular formation of columns with four rows. The amount of columns in the state is represented by Ns and is the same with the block length divided by 32.

Rijndael encryption algorithm has four steps in each round. And they are, SubBytes (byte substitution), ShiftRow, MixColumns, and AddRoundKey. Before the first round of transformation, the input block is processed by AddRoundKey. And again, the last round bypasses the MixColumns step. Or else, all rounds are the equal bar each one of them operates a separate round key, and the output of one round will be used as the input for the subsequent round of transformation. For the decryption aim, the mathematical inverse of each step is utilized, in reverse order; specific changes permit this to come out like the same steps as encryption with constants changed. Every single round key computation equally needs the SubBytes operation [25].

In the same vein, [1], in its description of Rijndael that the block and cipher key are often represented as an array of columns where each array has 4 rows and each column represents a single byte (8 bits). The number of columns in an array denoting the state or cipher key, then, can be computed as the block or key length divided by 32 (32 bits = 4 bytes). An array representing a State will have **Nb** columns, where **Nb** values of 4, 6, and 8 correspond to a 128-, 192-, and a 256-bit block, respectively. Similarly, an array representing a Cipher Key will have **Nk**columns, where **Nk** values of 4, 6, and 8 correspond to a 128-, 192-, and a 256-bit key, respectively. An example of a 128-bit State (**Nb**=4) and 192-bit Cipher Key (**Nk**=6) is shown below:

$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$
$s_{1,0}$	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$
$s_{2,0}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$
$s_{3,0}$	$s_{3,1}$	$s_{3,2}$	$s_{3,3}$

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$	$k_{0,4}$	$k_{0,5}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$	$k_{1,4}$	$k_{1,5}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$	$k_{2,4}$	$k_{2,5}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$	$k_{3,4}$	$k_{3,5}$

The number of transformation rounds (**Nr**) in Rijndael is a function of the block length and key length, and is given by the table below:

No. of Rounds Nr		**Block Size**		
		128 bits Nb = 4	**192 bits Nb = 6**	**256 bits Nb = 8**
Key Size	**128 bits Nk = 4**	10	12	14
	192 bits Nk = 6	12	12	14
	256 bits Nk = 8	14	14	14

The AES/Rijndael cipher itself has three operational stages:

- AddRound Key transformation
- **Nr**-1 Rounds comprising:
  - SubBytes transformation
  - ShiftRows transformation
  - MixColumns transformation
  - AddRoundKey transformation
- A final Round comprising:
  - SubBytes transformation
  - ShiftRows transformation
  - AddRoundKey transformation

Rijndael operates on a 4x4 column order of matrix of words (A word is equal to 2 bytes) also known as the state. (Zhang et al, 2005).

3 Methodology

This section would be explained using an analytical approach

Choice of design: Simplicity, protection against all familiar attacks, code compactness, and speed on a broad type of platform are key designs choices of Rijndael [1]. This design is hinged on security and efficiency which was realized through parallelism and modularity, symmetry, and lack of arithmetic operations which not only affects security but efficiency [19]. Rijndael is an iterated cipher with a variable block length and variable key length. Having a good design choice for the development of a secure system. [26].

Implementation: Rijndael has been implemented in many programming languages [11]. [9]. [12], implements Rijndael block ciphers in reconfigurable hardware based on sample digital design rules applied to iterated cipher block. Multiplexor model, RAM-based implementation, and composite field solution are different ways of implementing the Rijndael algorithm.

[13], implements Rijndael with Field Programmable Gate Arrays (FPGA) with reasonable efficiency and a quicker customizable solution. It can also be implemented on a wide range of processors and smart cards [15]

It is not only faster executed in both of the hardware and software, it is also the latest algorithm that is required by the United State and international standard and also more securely to use [16].

Rijndael is suitable to be executed in specialized hardware, though there is a various tradeoffs between chip area and speed [10]. Lots of hardware implementation of Rijndael encryption algorithm using VHDL is abounded. [17]. Rijndael can also be implemented on an 8-bit microcontroller [14]. Rijndael low memory requirement makes it a good choice for a limited space environment in which it shows good performance [9]). Various software implementations have been realized with assembly for greater performance.

Performance

The main basis for selecting Rijndael was its good performance over different environments from 8-bit smart cards to 32-bit computer processors [9]. Its performance depends on a particularly high level of language use and most of the time the software affects the performance.

According to [21]. Rijndael outperforms other symmetric algorithms like SERPENT, TWOFISH, RC6, etc. at the same time when we look at both numbers of request, processes per second. It also outperformed DES, 3DES, and RC2 [20] Also in terms of memory usage, Rijndael has an advantage over blowfish and RC6 [21].

Strength

Rijndael encryption algorithm has a simple elegant structure. It shows its structure by not using complex components rather it profit from the use of basic elements in a fine established structure [19]. It is regarded as maintaining a huge algebraic structure that makes the cipher's security to be simply determined with a short time which invariably provides a benefit over more complicated designs that demand vast thinking, searching, and bit tracing [19].

[16], Rijndael is the most used algorithm compared to other cipher algorithms. It also has the advantage of more secure executed communication when set side by side with other encryption algorithms.

Weakness

The choice of Rijndael as at the time it was adopted was no doubt a good selection but because of the recent advance in technology progress which now require a recalculation of the algorithm to be able to cater for the present and future [3].

The principal weakness of Rijndael is the issue of linearity in the S-box and the key schedule. Not only is Rijndael encryption too difficult to details its process, but it is also too patented which makes it complex to decrypt data if the private key is lost.

Round Transformation

Rijndael is an iterated cipher block, the encryption or decryption of a block of data is achieved by the iteration of a peculiar transformation. With a repeated block cipher, the different transformations work in sequence on intermediate cipher results i.e. states [19].

4. Conclusion

This paper presented a literature review and analytical study for the Rijndael algorithm and the main standard principles for the design principles. Rijndael choice of design in terms of simplicity, resistance against all familiar attacks, and capability to work on broad types of platforms was one of the reasons it was selected as the Advanced Encryption Standard (AES) among other contestants to replace the flawed insecure DES by National Institute Science and Technology (NIST). The basic target of this study is to highlight Rijndael weaknesses, vulnerabilities, and strengths. Its ability to be implemented on both software, hardware, low RAM, low power consumption constrained devices would no doubt be useful to improve the security of the Internet of Things (IoT).

5. Recommendation

Further research can be done to come up with An Enhanced Rijndael Algorithm can be made by integrating a suitable symmetric and asymmetric cryptography algorithm taking factors like low Ram and low power consumption into consideration that would be executed on the Internet of Things to further improve the security of the Internet of Things.

REFERENCES

Journal Papers:

[1] Tanzir Ismat(2015) *Comparative Analysis of AES Algorithms and Implementation of AES in Arduino* Thesis report. School of Engineering and Computer Science BRAC University. http://hdl.handle.net/10361/4889

[2] Thomas Gagne (2015) *Building an Algebraic Representation of the AES in Sage University of Puget Sound Sound Ideas SummerResearch.. Paper 243. Accessed online from http//sound ideas.pugetsound.pugetsound.edu*.

[3] Omar A Dawood & Othman I Hammandi (2017). An analytical study for some Drawbacks and weakness points of the AES cipher (Rijndael Algorithm. The 1st International Conference on Information Technology (ICOIT 17) Lebanese French University – Erbil Kurdistian Region – Iraq 10th of April 2017.

[4] Garima Saini & Naveen Sharma (2014) *Triple Security of Data in Cloud Computing Garima Saini et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4) , 2014, 5825-5827*

[5] Kohnstamm , J., & Madhub , D. (2014). Mauritious Declaration of The Internet of Things. *36th International Conference of Data Protection and Privacy commissioners*

[6] Evans, D. (2011*). The Internet of Things. How the Next Evolution of The Internet is Changing Everything. Accessed online from http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf on 04-05-2021*.

[7] Goldman Sachs, G. (2014). *The internet of Things. Making sense of the next mega trend IOT primer. Goldman Sachs Global Investment Research..*

[8] Gianmarco Baldini, Trevor Peirce, Marcus Handte, Domenico Rotondi, Sergio Gusmeroli, Salvatore Piccione, Betrand Copigneaux, Frank Le Gall, Foued Melakessou, Phillipe Smadja, Alexandru Sebanati, Julinda Stefa. (2013*): Internet of Things, Privacy Security and Governance; Internet of Things: Converging Technologies for smart Environments and integrated Ecosystems. 207-224W.J.*

[9] Daniel Fowls (2008): *The Implementation and Analysis of the Rijndael Encryption Algorithm in different programming language*: Bachelor of Science in Computer Science with honors. The University of BATH.

[10] Joan Daemen & Vincent Rijmen (2002) *The design of Rijndael AES. The advanced Encryption algorithim. Springer science and Business media.. http://www.springer.com/978-3540-42580-9*

[11] Joan Daemen & Vincent Rijmen (2002). *Rijndael Beyond AES. Mikulasska Kryptobesidka (2002).*

[12] Francois- Xavier Standeert Gael Rouvroyi, Jean Jacques Quisquaker,

Jean Didier Legat (2005). *Efficient Implementation of Rijndael Encryption in Reconfigurable Hardware : Improvements and Designs Tradeoffs.*

[13] Rajender Manteena (2004) *A Vhdl Implementation Of The Advanced Encryption Standard-Rijndael Algorithm* A thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Electrical Engineering Department of Electrical Engineering College of Engineering University of South Florida.

[14] Sungha Kim & Ingrid Verbauwhede (nd) AES Implementation On 8-Bit Microcontroller Department Of Electrical Engineering University Of California, Los Angeles Los Angeles, CA-90024.

[15] Sarah Merrion (2002) Rijndael – *The Future of Enc ryption Global Information Assurance Certification Paper Copyright SANS Institute.* http://www.giac.org/registration/gsec.

[16] Huong , M. H. (2014*). Implementation of AES Advanced Encryption Standard algorithm in communication application.* A thesis is to submit in partial to perform of the requirement for the award of the Degree of Bachelor Computer Science (Computer Systems & Networking) Faculty of Computer System & Software Engineering UNIVERSITI MALAYSIA PAHANG.

[17] L.Thulasimani & M.Madheswaran (2010*) Design And Implementation of Reconfigurable Rijndael Encryption Algorithms For Reconfigurable Mobile Terminals (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 04, 2010, 1003-1011*

[18] Harsh Kumar Verma & Ravindra Kumar Singh (2012*). Performance Analysis of RC6 TWOFISH and Rijndael Block Cipher Algorithms. International Journal of Computer Applications (0975-8887) Volume 42, No 16 March 2012.*

[19] Mukund S. Wankhade & Pravin D. Soni (2013). *Advanced Cryptanalytic Algorithm for Data Security. International Journal of Application on Innovation in Engineering & Management (IJAIEM) VOL 2, Issue 3, March 2013*.

[20] Abdel-Karim Al Tamini (2008) *Performnace Analysis of Data Encryption Algorithms.* http://www.cse.wustl.edu/~jain/cse567-06/encryption_perf.htm.
[21] *T. Christopher & Mohana Priya. A (2016) Study of Symmetric key Network Security Algorithms. International Journal for scientific Research and Development. Vol 3, issue 11, 2016.*

[22] Nitin K. Jharbade† and Rajesh Shrivastava(2012) *Network based Security model using Symmetric Key Cryptography (AES 256– Rijndael Algorithm) with Public Key Exchange Protocol (Diffie-Hellman Key Exchange Protocol) IJCSNS International Journal of Computer Science and Network Security, VOL.12 No.8, August 2012*.

[23] Micheal Cobb. https://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard*. Accessed online on 09-04- 2021*

[24] R. Velayutham & D. Manimegala (2010*). Analysis of AES Hardware and Software Implementation Oriental Journal of Computer Science and Technology Vol 3 (1) 83-88, 2010.*

[25] Shiyong Zhang, Gongliang Chan, La Fan, Jianhua Li (2005). *The Algorithm of AES. Accesed from http://citeseerx.ist.psu.edu/viewdoc/citations?doi=10.1.1.301.7140 on 11-05-2021*.

[26] Gary C. Kessler (2017) *An Overview of Cryptography. Accessed from http://www.garykessler.net/library/crypto.html on 16-06-2021.*

.