Journal of Engineering and Sustainable Development
http://jeasd.uomustansiriyah.edu.iq/index.php
https://doi.org/10.31272/jeasd.25.4.6

Vol. 25, No. 04, July 2021
ISSN 2520-0917

ENHANCING HYBRID SECURITY APPROACH USING AES AND RSA ALGORITHMS

*Samir G. Chaloop[1] Mahmood Z. Abdullah[1]

1) Computer Engineering Department, College of Engineering, Mustansiriyah University, Baghdad, Iraq.

Received 11/12/2020 ***Accepted in revised form 17/1/2021*** ***Published 1/7/2021***

Abstract: Network safety has become an important issue in recent years. Encryption has been developed as a solution and plays an important role in the security of information systems. Many methods are required to secure the shared data. The advanced internet, networking firms, health information and the cloud applications have significantly increased our data every minute. The current work focuses on cryptography to provide the protection for sensitive data that exchanged between personal users, companies, organizations, or in the cloud applications and others during the transfer of data across the network. Firstly, Data sent from sender to network receiver must be encrypted using the cryptographic algorithm. Secondly, the recipient shows the original data using the decryption technique. This paper presents three encrypting algorithms such as AES, RSA and hybrid algorithms, and their efficiency is compared based on the analysis of the time. Results of the experiments show that the hybrid algorithm is better in term of security.

Keywords: *AES algorithm*, *Hybrid algorithm*, *OAEP*, *RSA algorithm*, *Scrypt*, *Security*.

1. Introduction

Due to the development in the information technology sciences and the importance of data, this has led to the need to secure this data in several ways. Insecure data is considered one of the problems at the present time with the development of information technology and the use of Internet networks through data correspondence between various parties, organizations and users. The term security of information refers to a methodology employed to satisfy the security requirements [1]. Encryption is one way to ensure the secrecy and access of the data to the receiving party without affecting it from any third party. More broadly, cryptography includes the creation and study of algorithms that prevent third party or public reading of private messages [2]. There are two forms of cryptographic algorithms. The first one is a public cryptography (PKC), or an asymmetric cipher that uses two keys, one to encrypt and one to decrypt by sender and receiver, respectively. The second is a symmetric cipher where the sender and the receiver use the same key for encryption and decryption. Encrypted data stays secret by not revealing the user's confidentiality [3]. In this article, a hybrid encryption algorithm is designed and built based on (AES + RSA) encryption algorithm. A hybrid cryptography contains multiple ciphers types that are usually based on different strengths. The idea is to generate a special encryption key and then use

**Corresponding Author*: samir89ghaly@gmail.com

the participant's public key to encode it. The encrypted key is sent to the receiver with the cipher text to protect important original data and ensure that the data is not known from any external attack until the recipient is received these data. This algorithm is implemented in a way that enables it to deal with any size of data, in addition to making some improvements that is increasing the level of protection for this data. The structured of remainder paper as follows: The contribution is mentioned in Section 2, Section 3 demonstrates the previous work, Section 4 presents the methodology of AES, RSA and hybrid implementation, Section 5 presents the results of the simulation, Section 6 presents the conclusion, and at last, in Section 7 the references are submitted.

2. Contribution

This article is an extension of the work presented by Lin Zou et al. [4]. This work optimizes the following:

- Less processing time for encryption and decryption processes by AES and RSA algorithms.
- Appending the AES and RSA algorithms by Scrypt and OAEP algorithms, respectively to prevent the keys from being broken by any external attacker.
- Increasing the throughput for cryptography algorithm.
- Building a hybrid algorithm to increase data security.

3. Related Works

(Dr. M. Tanooj kumar et al., 2019) [5] Proposed a hybrid HAR (Hybrid AES Rail Fence) encryption algorithm for safe data communication through IoT devices. A hybrid encryption solution, combines the AES and Rail Fence, enhances protection in different real-time applications such as securing passwords on account, securing secret word message for listed bank account users (OTP) and many other applications. The suggested encryption algorithm guarantees secrecy so that attackers are unable to read the cipher text. This model (HAR) offers greater protection in terms of data search complexity and improves results compared with AES algorithm.

(Lin Zou et al., 2020) [4] Proposed the use of AES and RSA hybrid encryption algorithms in file encryption, they introduced the basic concepts of AES and RSA algorithms to ensure the protection of personal or enterprise data, and evaluated their advantages and disadvantages. The study showed that the hybrid encryption algorithm optimized the data security, key management and efficiency.

(Dwi Yuny SYLFANIA et al., 2020) [6] Proposed Blowfish and RSA algorithms for email transmission and reception based on Android basis. The algorithms have been evaluated and compared to determine which algorithm has an excellent speed in encryption and decryption time. The consequence of this research is that the Blowfish algorithm is faster than the RSA algorithm. It was demonstrated from the test results that Blowfish was 178.958 percent faster than RSA for the encryption process. RSA is 63.131 percent slower compared to Blowfish. The same result is obtained for the decryption process, Blowfish, the fastest compared to RSA, by 420.44188 percent. Instead, RSA is 80.3399 percent slower than Blowfish.

(Samar Zaineldeen and Abdelrahim Ate, 2020) [7] They proposed a new hybrid encryption algorithm composed of AES and EHC algorithms in order to ensure the safe communications between the user and the server side. Compared to AES-128 and DES algorithms, the proposed method can provide enhanced results in terms of security, throughput, memory consumption and time for encryption and decryption. The testing showed that the proposed algorithm takes the least amount of time to encrypt and decrypt data, and that memory usage is more effective with the processes.

4. Methodology

4.1. AES Algorithm

AES is the block cipher, where the NIST has chosen under the name Rijndael. The block cipher of Rijndael works with the SPN system and encrypts a 128-bit fixed block of data with three cipher keys under the NIST specifications. The algorithm's condition consists of 16 bytes, typically a matrix of 4*4 [8]. The rounds number of AES algorithm is based on the key length, as shown in Table 1.

Table 1. Rounds number according to key size

Key size	Rounds number
128-bit	10 Rounds
192-bit	12 Rounds
256-bit	14 Rounds

Maximum key size used in this work (256 bit). Each round of AES encryption function consists of four different transformations [9]:

4.1.1. Byte substitution Transformation

The Byte substitution transition is a non-linear replacement, which performs autonomously on each state byte. The transformation of Sub Byte is performed using the once pre-calculated replacement table named the S-box. This table includes 256 (0-255) numbers and their performance.

4.1.2. Shift Row Transformation

In the shift row transformation, the state rows are left cyclically over multiple offsets. Row zero is not relocated; row one is moved by one byte; row 2 is moved by 2 bytes; three bytes are shifted to row 3.

4.1.3. Mix Column Transformation

It works independently on each column. It takes all of the state columns and combines the data to create new columns.

4.1.4. Add Key Transformation

The result round key from the combining column transformation process is implemented in the state with a simple X-OR bitwise. The circular key is equal to the block size. Any round key consists of the key timeline NB words. Each NB word is applied to a state column. The output of the transformations above is called 'state.' In each message block the state has the same length of bytes.

The AES key is generated by using Scrypt with GCM mode. Scrypt is a computationally, intensive main derivative function that requires more time to measure. This functionality provides users with a high level of protection, making it difficult for intruders to break the original passwords. The minimum length of the alphanumeric password is 8 characters and is then entered as the input for a Scrypt Hash function. The result of the hash value is known as a 256-bit key (s-key) [10]. The block diagram of the Scrypt algorithm is shown in Figure 1.

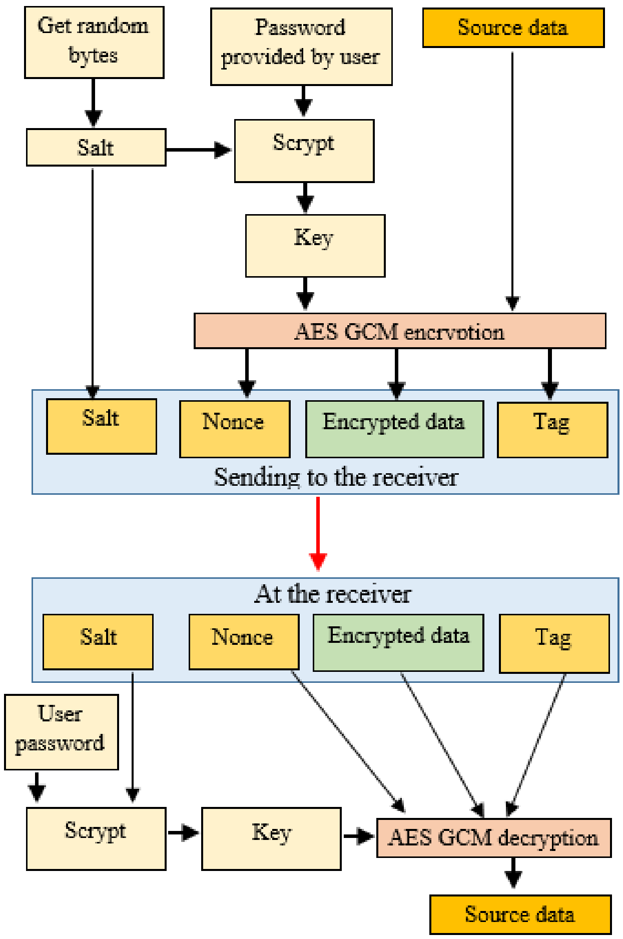

Figure 1. AES with Scrypt algorithms

The key schedule is the same with AES decryption; the only steps that we need to do are the inverse operations of encryption, and the same add round key [11].

4.2. RSA Algorithm

RSA is an asymmetric algorithm for encryption. There are two keys, one to encrypt messages with an (all) public key and the other to decrypt them with a private key (secret, known to the user only). Encrypt a plaintext using a public key that can only be decrypted using a private key only. A public key can be accessed by all persons via the device communication mode and stored as a pke file. The private key is protected by saving it as a kez file and not transferred anywhere. The sender's public and private keys help the user prove the correctness of the data [12]. Three major steps of the RSA algorithm; Key generation, in addition to encryption and decryption. The largest key size is used in this work (4096 bit) supported by OAEP. OAEP algorithm is a pad bit generation. This algorithm is a popular algorithm used with RSA for secure encryption. The pad bits are created randomly to make it difficult for the attacker to handle either the plaintext or the pad bits [13]. The RSA algorithm flowchart, as shown in Figures 2(a) and 2(b).

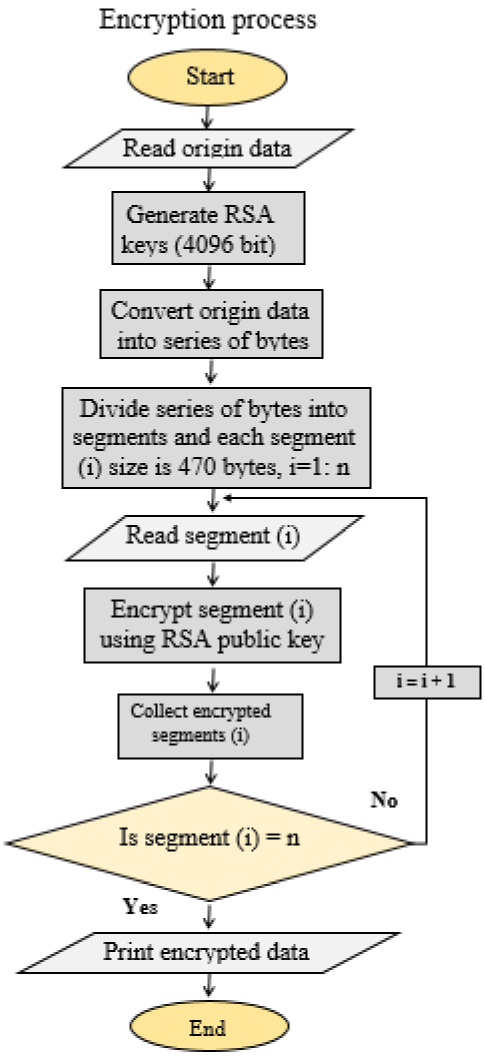
Encryption process
Start
Read origin data
Generate RSA keys (4096 bit)
Convert origin data into series of bytes
Divide series of bytes into segments and each segment (i) size is 470 bytes, i=1: n
Read segment (i)
Encrypt segment (i) using RSA public key
i=i+1
Collect encrypted segments (i)
No
Is segment (i) = n
Yes
Print encrypted data
End

Figure 2(a). The RSA encryption algorithm flowchart

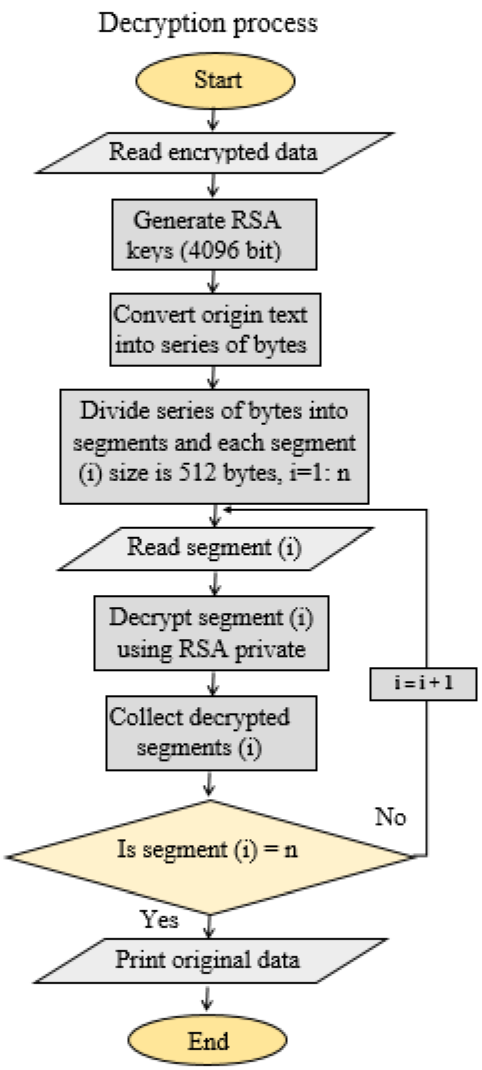
Decryption process
Start
Read encrypted data
Generate RSA keys (4096 bit)
Convert origin text into series of bytes
Divide series of bytes into segments and each segment (i) size is 512 bytes, i=1: n
Read segment (i)
Decrypt segment (i) using RSA private
i=i+1
Collect decrypted segments (i)
No
Is segment (i) = n
Yes
Print original data
End

Figure 2(b). The RSA decryption algorithm flowchart

4.3. Hybrid Algorithm

In this paper, presents a hybrid encryption method to protect data transfer by two encryption techniques: AES algorithm used to encrypt the text files; asymmetric RSA algorithm used to encrypt AES key to ensure a safe transmission between client-client or client-server from verifying by other person, to make it more difficult to access by the attacker. In this algorithm, the original data enters the system, and the output is the encrypted data by AES key with encrypted AES key by RSA public key during one execution stage and not in separate stages. At the receiver, the decryption process is done. Decoding is the opposite of encoding processes, by which the original text is obtained. The block diagram of the hybrid algorithm, as shown in Figure 3.

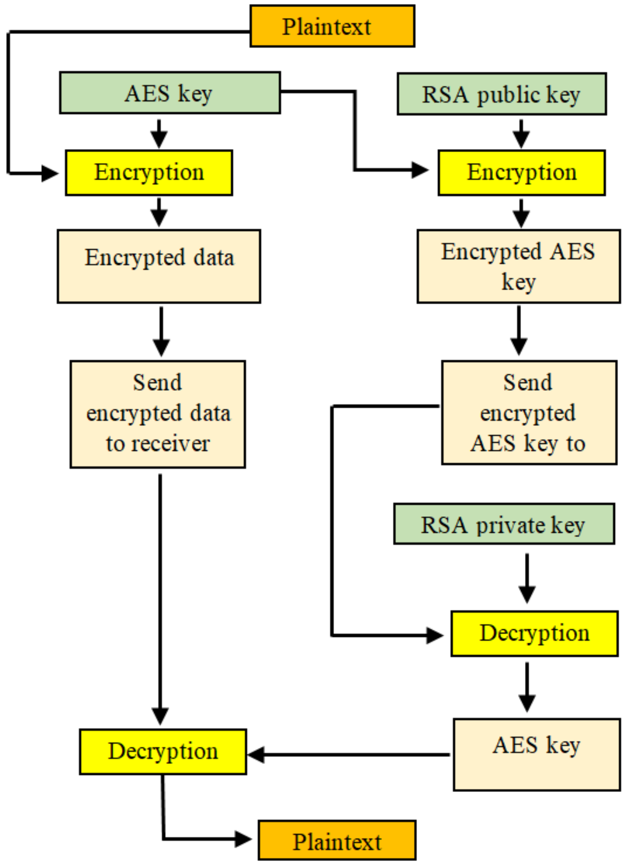

Figure 3. The block diagram of hybrid algorithm

5. Experiments and Results

A code of Python 3.8 has been written to implement the proposed cryptography algorithm with Windows 10 (64 bit) operating system. HARD 320 GB, RAM 4 GB DDR3, CPU Intel® coreTM i3-2310M are computer specifications. Using fourteen text files with different sizes (1.19 MB, 3.57 MB, 7.14 MB, 10.7 MB, 17.8 MB, 21.4 MB, 25 MB, 28.5 MB, 32.1 MB, 35.7 MB, 39.2 MB, 42.8 MB, 46.4 MB and 50 MB) in this paper. Calculating the encryption and decryption time, and compared all the algorithms results. Several approaches have been defined as follows:

5.1. Data security

Three algorithms are used in this experiment to encrypt and decrypt the same text files. Results show that the best algorithm is the hybrid algorithm which provides the reliability and the highest security for the data sent when the RSA,

AES and hybrid algorithms are compared, the encryption and decryption time differences evaluated in Figure 4. The encryption-decryption schedule of the three is shown in Table 2 with different files size. Table 3 shows the total processing time in second. According to the results shown in the table 3, it is clear that the hybrid algorithm is better than the other algorithms (AES and RSA) after testing the three algorithms and calculating the time required to complete the processing on several text files of different sizes.

Table 2. Time taken for each algorithm in second

Files size (MB)	RSA encryption	RSA decryption	AES encryption	AES decryption	Hybrid encryption	Hybrid decryption
1.19	0.04	0.60	0.015	0.03	0.025	0.51
3.57	0.11	0.86	0.06	0.07	0.07	0.55
7.14	0.21	1.31	0.14	0.17	0.15	0.65
10.7	0.31	1.76	0.18	0.21	0.19	0.69
17.8	0.49	2.68	0.31	0.32	0.32	0.81
21.4	0.59	3.2	0.37	0.41	0.38	0.88
25	0.70	3.73	0.45	0.49	0.46	0.97
28.5	0.79	4.62	0.53	0.62	0.54	1.10
32.1	0.91	4.73	0.601	0.67	0.61	1.15
35.7	0.98	5.26	0.65	0.68	0.66	1.16
39.2	1.07	5.76	0.71	0.76	0.71	1.24
42.8	1.18	6.26	0.82	0.84	0.83	1.32
46.4	1.28	7.04	0.92	1.04	0.93	1.52
50	1.37	7.28	1.09	1.18	1.12	1.66

Table 3. Total time for each algorithm in second

Files size (MB)	RSA total time	AES total time	Hybrid total time
1.19	0.64	0.045	0.535
3.57	0.97	0.13	0.62
7.14	1.52	0.31	0.8
10.7	2.07	0.39	0.88
17.8	3.17	0.63	1.13
21.4	3.79	0.78	1.26
25	4.43	0.94	1.43
28.5	5.41	1.15	1.64
32.1	5.64	1.271	1.76
35.7	6.24	1.33	1.82
39.2	6.83	1.47	1.95
42.8	7.44	1.66	2.15
46.4	8.32	1.96	2.45
50	8.65	2.27	2.78

The hybrid encryption algorithm provides the highest level of data security than other algorithms, and its speed is much faster than RSA and slightly slower than AES algorithms.

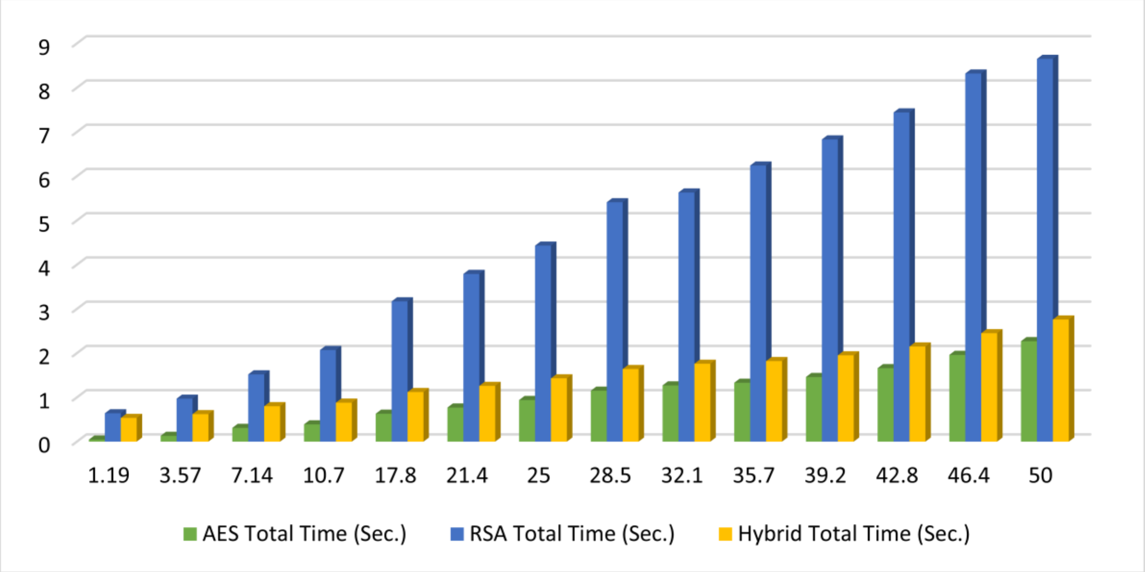

Figure 4. The encryption and decryption total time of the three algorithms

5.2. Throughput

Throughput is described as the amount of data that passes via a network system. It is a result of dividing all data send in Mega Byte by the average time required to send all data in second. Throughput value in (MB/Sec) for each algorithm as shown in Table 4. Throughput analysis is shown in the Figure 5.

Table 4. Throughput value of the algorithms

RSA throughput	Hybrid throughput	AES throughput
77.73	239	353.54

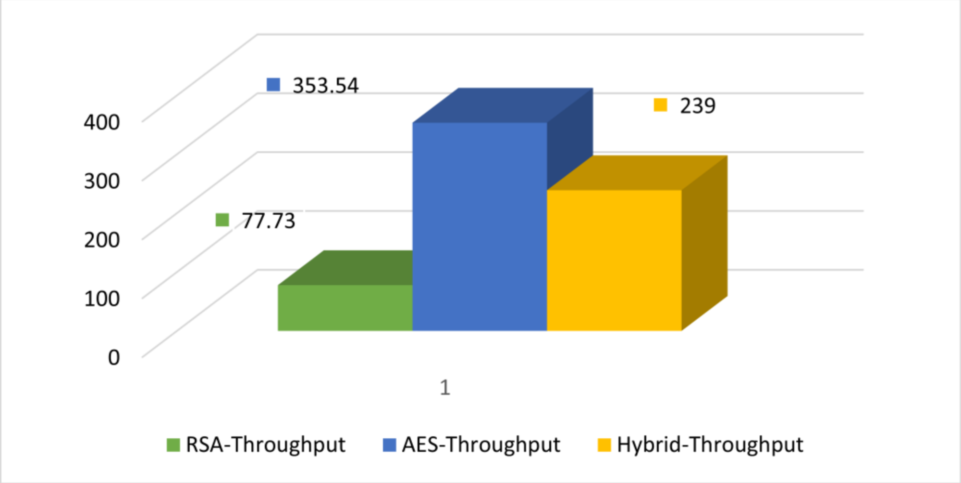

Figure 5. Throughput value of the algorithms

6. Conclusion

This work blends two kinds of encryption to take advantage of each one to create a hybrid algorithm with a high-security of data. AES is used to encrypt transmitted data. RSA is used to encrypt the AES key using public key. The encryption key and data are forwarding to the recipient, and the encrypted key is decrypted using the private key of RSA algorithm to get the AES key, and then the AES key used to decrypt the encrypted data to get the original text.

According to the experiments that applied in this paper, the algorithm of the hybrid encryption can be used in software application, system design, and other fields that required to exchange data security, which can effectively protect the data, in addition to the performance and fast execution time, as the results showed that the hybrid encryption is 67.47% faster than the RSA algorithm and 32.39% slower than AES algorithm.

Conflict of interest

The authors announce that the publishing of this article does not give rise to any conflict of interest.

7. References

1. Oleiwi, Z. C., Alawsi, W. A., Alisawi, W. C., Alfoudi, A. S. and Alfarhani, L. H., (2020). "*Overview and Performance Analysis of Encryption Algorithms*," J. Phys. Conf. Ser., vol. 1664, no. 1, doi: 10.1088/1742-6596/1664/1/012051.
2. Chinnasamy, P., Padmavathi, S., Swathy, R. and Rakesh, S., (2021). "*Efficient data security using hybrid cryptography on cloud computing*," Lect. Notes Networks Syst., vol. 145, no. September, pp. 537–547, doi: 10.1007/978-981-15-7345-3_46.
3. Banasode, P. and Padmannavar, S., (2018). "*Protecting and Securing Sensitive Data in a Big Data Using Encryption*". EAI Endorsed Trans. Smart Cities, vol. 0, no. 0, p. 163991.
4. Zou, L., Ni, M., Huang, Y., Shi, W. and Li, X, (2020). "*Hybrid encryption algorithm based on AES and RSA in file encryption"*. Springer volume 551 https://doi.org/10.1007/978-981-15-3250-4_68
5. Kumar, M. T., Katragadda, R. K., Kolli, V. S. and Rahiman, S. L., (2019). "*A hybrid approach for enhancing security in internet of things (IoT)*". Proc. Int. Conf. Intell. Sustain. Syst. ICISS 2019, pp. 110–114.
6. SYLFANIA, D. Y., JUNIAWAN, F. P., LAURENTINUS and PRADANA, H. A., (2020). "*Blowfish–RSA Comparison Analysis of the Encrypt Decrypt Process in Android-Based Email Application".* vol. 172, pp. 113–119.
7. Zaineldeen, S. and Ate, A., (2020). "*Improved cloud data transfer security using hybrid encryption algorithm*," Indones. J. Electr. Eng. Comput. Sci., vol. 20, no. 1, pp. 521–527, 2020, doi: 10.11591/ijeecs. v20.i1. pp. 521-527.
8. Su, N., Zhang, Y. and Li, M., (2019). "*Research on data encryption standard based on AES algorithm in internet of things environment*". Proc. 2019 IEEE 3rd Inf. Technol. Networking, Electron. Autom. Control Conf. ITNEC, pp. 2071–2075.
9. For, C., Technology, I., Ramkumar, G. V., For, C. and Technology, I., "*ADVANCED DATA SECURITY IN CLOUD COMPUTING BY AES AND DES HYBIRD Text Image Audio*".
10. Bidhuri, V., Heffernan, N. and Heffernan, N., "*Enhancing Password Security Using a Hybrid Approach of SCrypt Hashing and*

AES Encryption”. MSc Internship Cyber Security National College of Ireland Supervisor.

11. G, P. V., (2019). “*AES Based Algorithm for Image Encryption and Decryption*”. vol. 2, no. 11, pp. 342–345.
12. Al-Kadei, F. H. M. S., Mardan, H. A. and Minas, N. A., (2020). “*Speed Up Image Encryption by Using RSA Algorithm*”. 2020 6th Int. Conf. Adv. Comput. Commun. Syst. ICACCS, pp. 1302–1307.
13. Mathematics, A., (2017). “*ijpam.eu*”. vol. 115, no. 6, pp. 689–695.