© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 13 Issue: 02 | April - June 2022
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed

Enhancing Data Storage Security in Cloud Using Cryptography

Ms. Shivani B. Nimje
Computer Science and Technology
St. Vincent Pallotti College of engineering and technology, Nagpur, India.

Prof. M. B. Gudadhe
Computer Science and Technology
St. Vincent Pallotti College of engineering and technology, Nagpur, India.

Abstract - In the field of cloud computing, data security is of paramount importance. For safe data storage, a variety of encryption algorithms are available, each with its own set of benefits and drawbacks. In identity-based encryption, there is a difficulty with the key escrow and certificate revocation. Personality-based encryption does not require a secure intermediary. The key escrow and certificate revocation issues will be solved using the certificate-less encryption technique. In certificate less encryption, the responsibility of key generation is split between the cloud and the client. The Blowfish and RSA algorithms will be implemented to establish whether the algorithm has outstanding speed, both algorithms will be analyzed and compared in terms of encryption and decryption speed. When compared to alternative symmetric algorithms, the Blowfish algorithm was chosen because of its better speed. The RSA algorithm is chosen because it is faster than the other asymmetric algorithms. Calculate how fast the techniques' encryption and decryption processes are with the identical key length and message characters. Time is measured ten times, then the average figure is taken to produce a consistent result. As our data input environment in Python, we used characters, and based on that, we came to some conclusions. The result of this study is that the blowfish method is faster than RSA when it comes to encrypting or decrypting data. Methods of encryption and decryption will be used to hide and reveal the number of characters.

Keywords: - Blowfish, RSA, Cryptography, Encryption, Decryption

1. INTRODUCTION

Computing in the cloud refers to a system that utilizes a shared resource pool and can be organized in a variety of different ways. These assets are utilized to provide access that is both ubiquitous and on-demand, and they can be immediately set up or taken down without requiring a significant amount of effort from the organization. The phrase "cloud computing paradigm" is frequently used to refer to cloud computing. It makes use of services, hardware, software, and infrastructure that are all accessible through the use of a network. Customers have access to tools for communication and transportation, as well as programming and system administration, thanks to the internet, which is used to facilitate this process. These levels disguise the complexity and intricacies of the underlying architecture by providing clients and applications with an extremely straightforward graphical interface or application programming interface (API). In addition, this structure provides interest benefits that can be relied on regardless of where, when, or how they are realized. Hardware and software services are made available to the general public, enterprises, corporations, and businesses on a pay-per-use and as-needed basis.

A method of encrypting communications in which both the sender and the recipient of a message have access to the same, singular key. This key is used for both encodings and decoding the communication. To counteract this, public key cryptology uses two keys: a public key to encrypt messages and a private key to decode them. The public key is used to encrypt messages, while the private key is used to decode them.

The symmetric-key algorithms use substantially less processing power than their asymmetric counterparts do. In actual use, asymmetric key calculations take significantly more time than their symmetric counterparts do. This is since with symmetric key calculations, we may make use of key match for encryption (public key) and decryption (private key). The asymmetric method, despite its slower processing speed, produces satisfactory results in terms of safely storing and transferring data. This is true even though the algorithm itself is inherently flawed. One of the responsibilities of symmetric key calculations is the requirement of a common secret key, with the two parties each keeping a similar copy at their respective ends. It is necessary to have a total of n(n - 1)/2 keys in order to ensure that everyone in a group of n people may communicate safely with one another. This is because there are an infinite number of channels through which communication can take place [1].

2. LITERATURE REVIEW

Ali E. Taki El_Deen found Comparison between AES, DES, RSA, and Blowfish encryption algorithms is discussed, and Statistical tests of AES, DES, RSA, and Blowfish algorithms have been examined.[Hybrid Encryption Algorithm combines the difficulty of estimating the original text and verity of using the different keys on symmetric and asymmetric encryption that introduce ciphertext more difficult for estimation so that algorithm is at a high level of security.[1]

Bello Alhaji Buhari, Afolayan Ayodele Obiniyi, Kissinger Sunday, Sirajo Shehu has Different sizes of data blocks, and standard key size is used to evaluate the algorithm's run time speed. All the implementations were exact to make sure that the results will be relatively fair and accurate. The data block sizes that are used in this research are 128kb for both AES and Blowfish.[2]

Mayes M. Hoobi implemented the security of any type of algorithm depending on the secrecy of the key. Based on the results in this research, the main conclusions are improved key functions increase the complexity in a block cipher, in addition to increase the search space of DES key (increase the probability of brute force attack that is used to cryptanalyze the cipher).[3]

S.V.N.Srivalli, Ben Swarup Medikonda has a paper introduces a modern approach which will help the user to reduce risk of security while to the text files. The storing process and the retrieval process in which the user needs to perform to which will reduce the risk of the security while transferring the text files. [4]

Fortine Mata, Michael Kimwele, George Okeyo found AES is the best symmetric encryption algorithm, it's more secure than Blowfish though compared to other algorithms Blowfish is by far the best. Blowfish gives the highest throughput as compared to AES. [5]

Wafaa A. N. A. AL-Nbhany, Ammar Zahary implemented Blowfish encryption and decryption is faster than AES and RSA because it takes the lowest processing time, and AES algorithm is faster than RSA algorithm in terms of encryption and decryption speed. [6]

Zijing Jiang , et.al [7] A symmetric encryption algorithm's most significant component is the S-box. Using chaos theory, many strategies are proposed. Create an S-box with good cryptological features by building a collection of Bent functions as the output. After that, the output bits' nonlinearity, differential uniformity, rigorous avalanche criterion, and independence criterion are evaluated and tested. The suggested S-box offers outstanding cryptographic features, according to a security study.

Hyunji Kim , et.al [8] The late fusion approach extracts two characteristics from a single-source data set, learns through each network, then combines them. Training functions in binary files correctly classify generic software and block cipher algorithms. The suggested technique not only detected crypto-ransomware with a 97 percent success rate but also classified each lightweight cryptographic algorithm and benign firmware with an 80 percent success rate.

2.1 BLOWFISH ALGORITHM – Symmetric Algorithm

In 1993 Bruce Schneier, one of the world's leading cryptologists, designed the Blowfish algorithm and made it available in the public domain, blowfish is a variable-length key, blowfish is also a block cipher with a length of 64 bit, and has not been cracked yet, it can be used in hardware applications due to its compactness [2][4][5]. There are two parts to this algorithm; a part that addresses the expansion of the key and a part that addresses the encryption of the data.

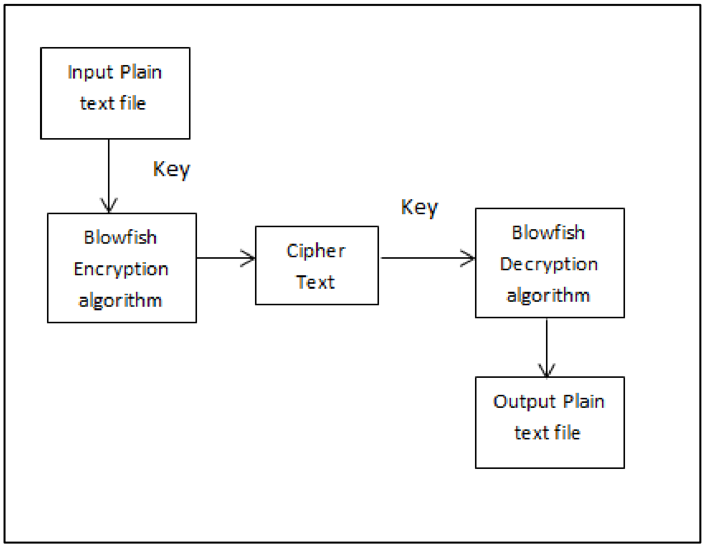


Fig 1: Blowfish Algorithm encryption and Decryption Process

Key Expansion

The key Expansion of the blowfish algorithm begins with the P-array and S-boxes with the utilization of many sub-keys, which requires pre-computation before data encryption or decryption. The array consists of eighteen 4-byte sub-keys: P1, P2…P17, P18.

Blowfish with keys up to 448 bits in length is transformed into several sub-key arrays.

There are 256 entries for each of the four 32-bit S-boxes:
S1, 0, S1, 1...., S1, 255
S2, 0, S2, 1...., S2, 255
S3, 0, S3, 1...., S3, 255
S4, 0, S4, 1...., S4, 255
Below are the steps of how to generate the subkeys:

- Initialize the P-array and four S-boxes with a fixed string; this string consists of hexadecimal digits of π.
- The first element in P-array (P1) is XORed with the first 32 bits of the key, and the second element in P-array (P2) is XORed with the second 32 bits of the key, repeated this until all the elements in P-array are XORed with the key bits.
- Encrypt all-zero string by blowfish algorithm using sub keys described in steps (1, 2).
- Change P1 and P2 with the output of step (3).
- Using the modified sub keys encrypt the output of step (3).
- Change P3 and P4 with the output of step (5).

This process is continued, until the entire, the P-array and 4 s-boxes are changed [4].

Having more requirements, though, does make it easier to encounter difficulties. The information that is saved on the computer can be hacked in a variety of different ways. A significant number of individuals currently save at least some of their private data and possibly even their anchored data in the cloud. Therefore, the most crucial issue to be concerned about with cloud computing is security.

Encrypting anything can be done in a variety of different ways. A straightforward type of ID-based cryptography is known as identity-based encryption, which is often referred to as identity-based encryption (IBE). The identification of the client, such as the client's email address, can be utilized as a public key within the context of encryption using a public key. This indicates that a sender who communicates with individuals who fall within the broad parameters of the system can encrypt a message utilizing a key such as the content value of the collector's name or email address. An authorized administrator is the one who provides the receiver with the decryption key. This administrator is reliable because he or she generates a unique secret key for each customer.

2.2 RSA ALGORITHM – Asymmetric Algorithm

The RSA algorithm is named after Ron Rivest, Adi Shamir, and Leonard Adleman. The public-key cryptography that was made possible by this algorithm was foundational to the ecommerce revolution that followed.

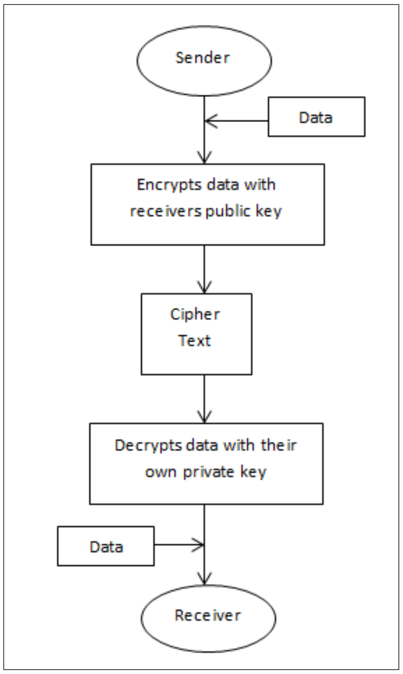

Sender
Data
Encrypts data with receivers public key
Cipher Text
Decrypts data with their own private key
Data
Receiver

Fig 2: RSA Algorithm encryption and Decryption Process

The RSA algorithm enables the following:

- Generate a random private key and public key to use in the key pair generating process (the size is 10244096 bits).
- Encryption: It takes a secret message (an integer in the range [0...key length]) and encrypts it with the public key, and then it takes that encrypted message and decrypts it using the secret key.
- Signing messages (using the private key) and verifying message signatures are both part of the digital signature process (using the public key).
- Key exchange: It ensures the safe transportation of a confidential key that is required for encrypted communication.

The starting point for learning the RSA algorithm is Euler's Theorem that was presented in Section 11.4 of Lecture 11. To recap, that theorem states that for every positive integer n and every a that is coprime to n, the following must be true

$$a^{\emptyset(n)} \equiv 1(mod\ n)$$

Where, as defined , $\emptyset(n)$ is the totient of n.

An immediate consequence of this theorem is that, when a and n are relatively prime, the exponents will behave modulo the totient $\emptyset(n)$ in exponentiated forms like a^k mod n. That is, if a and n are relatively prime, the following must be true for some k^1 and k^2:

$$a_k \equiv a_{k1\cdot\emptyset(n)+k2} \equiv a_{k1\cdot\emptyset(n)}a_{k2} \equiv a_{k2}\ (mod\ n)$$

For example, consider $a = 4$ in arithmetic modulo 15. The totient of 15 is 8. (Since $15 = 3 \times 5$, we have $\emptyset(15) = 2 \times 4 = 8.$) You can easily verify the following:

$$4^{7} \cdot 4^{4}\ mod\ 15 = 4^{(7+4)8}\ mod\ 15$$

$$= 4^{3}\ mod\ 15$$

$$= 64\ mod\ 15$$

$$= 4$$

$$(4_{3})_{5}\ mod\ 15 = 4_{(3\times5)8}\ mod\ 15$$

$$= 4^{7}\ mod\ 15 = 4$$

Note that in both cases the base of the exponent, 4, is coprime to the modulus 15.

The relationship shown above has some incredible ramifications that point to practical possibilities: To see what I mean, say that M is an integer that represents a message (note that any bit string in the memory of a computer represents some integer, no matter how large). Let's now conjure up two integers e and d that are each other's multiplicative inverses modulo the totient $\emptyset(n)$. Assume again that M is coprime to the modulus n. Since the exponents of M are going to behave modulo the totient $\emptyset(n)$, the following must be true.

$$M_{e\times d} \equiv M_{e\times d\ (mod\ \varphi(n))} \equiv M\ (mod\ n)$$

The result shown above, which follows directly from Euler's theorem, requires that M and n be coprime. However, as will be shown in Section 12.2.3, when n is a product of two primes p and q, this result applies to all $M, 0 \leq M < n$. In what follows, let's now see how this idea can be used for message encryption and decryption.

3. SYSTEM METHODOLOGY

As defined in fig 1, the overall system is proposed to work in python environment. In this environment we have created our algorithm function form blowfish and RSA to work with proper GUI. The input is the text files in character which are encrypted or decrypted as per the requirement of the user. The text file gets generated in blowfish and in RSA the string encryption is generated.

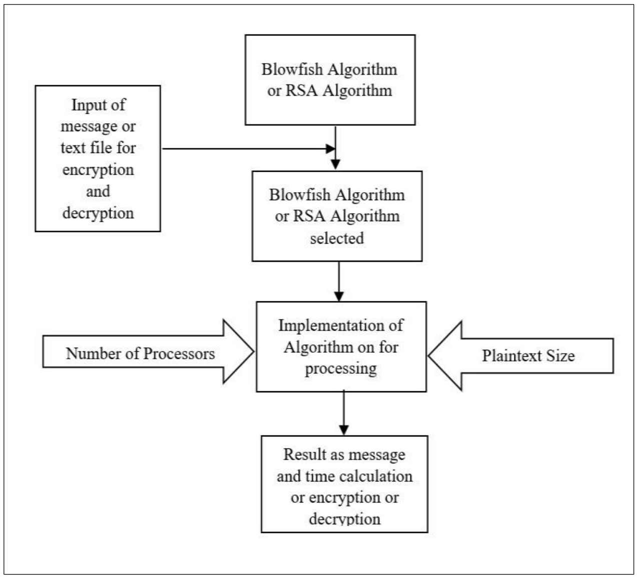
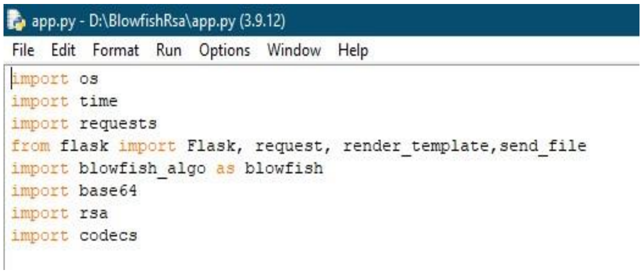

Blowfish Algorithm or RSA Algorithm
Input of message or text file for encryption and decryption
Blowfish Algorithm or RSA Algorithm selected
Number of Processors
Implementation of Algorithm on for processing
Plaintext Size
Result as message and time calculation or encryption or decryption

Fig 3: Showing the overall methodology of the work.

3. ENVIRONMENT

3.1 Python

The python environment has major 6 libraries which are important for the implementation of this research work as shown in fig 4.


app.py - D:\BlowfishRsa\app.py (3.9.12)
File Edit Format Run Options Window Help

```
import os
import time
import requests
from flask import Flask, request, render_template,send_file
import blowfish_algo as blowfish
import base64
import rsa
import codecs
```

Fig 4: Showing the libraries which are important for the work.

Blowfish is a symmetric block cipher designed by Bruce Schneier. It has a fixed data block size of 8 bytes and its keys can vary in length from 32 to 448 bits (4 to 56 bytes). Blowfish is deemed secure and it is fast. However, its keys should be chosen to be big enough to withstand a brute force attack (e.g. at least 16 bytes).

PyCryptodome libraries

PyCryptodome is an incredible library in python. Python has a wide variety of libraries available for encrypting and decrypting messages. PyCryptodome is a self-contained

Python package of low-level cryptographic primitives that supports Python 2.6 and 2.7, Python 3.4 and newer, and. PyCryptodome has been enhanced to add more implementations and fixes to the original PyCrypto library.

SQLite 3

SQLite3 is a very easy to use database engine. It is self-contained, serverless, zero-configuration and transactional. It is very fast and lightweight, and the entire database is stored in a single disk file. It is used in a lot of applications as internal data storage.

Google Colab

Colab notebooks execute code on Google's cloud servers, leverage the power of Google hardware, including TPUs and GPUs, regardless of the power of your machine.

3.2. SOFTWARE TESTING

The speed of encryption and decryption of Blowfish and RSA in a message with the same number of characters is compared. Graphs depicting the outcomes of this test. The user must first choose between the Blowfish and the RSA algorithms then provide plain text and a key. The plain text became cypher text after that process was completed. The decryption process follows; at this point, the cipher text is available in the menu inbox. Users pick which messages should be decrypted using the Blowfish or RSA algorithms, and then provide the key for that algorithm. The cipher text will be converted to plain text after the operation is completed.

4. RESULTS

This work encrypts and decrypts communications using two different algorithms, most likely Blowfish and RSA. The results of the encryption test and the results of the decryption test are divided into two sections in the discussion section, with the performance results of the two methods in each segment.

4.1 TESTING OF ENCRYPTION

The software testing result was displayed in a graph that depicted how long the encryption and decryption processes took based on the number of characters in the message. The encryption testing chart is presented in Table 1 compares the speed encryption of blowfish and RSA, as well as RSA to Blowfish

Characters	Blowfish (ms)	RSA (ms)	Blowfish against RSA percentage (%)	RSA against Blowfish percentage (%)
10	1	3	200	66.66667
50	2	7	250	71.42857
100	3	12	300	75
200	6	18	200	66.66667
Average			237.5	69.94048

Table 1: Compares the speed encryption of blowfish and RSA

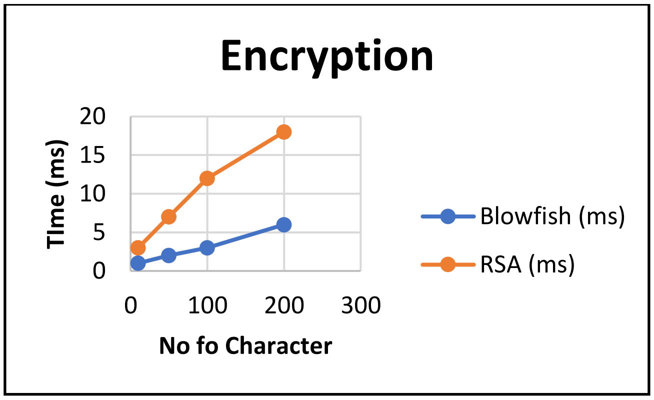


Fig 5: Graphical chart for Encryption

The length encryption procedure for Blowfish and RSA algorithms is described in the graph in Figure 6. The x-axis represents the number of characters to be encrypted, while the y-axis represents the encryption time in milliseconds. The testing is done ten times for each number of characters, and the figures in the chart are averages. Blowfish is 237.5 percent faster than RSA on average; RSA, on the other hand, is 69.94 percent slower. The Blowfish encryption algorithm requires substantially less time than the RSA algorithm, as seen in Figure 5 and Table 1. As a result, the Blowfish encryption algorithm outperforms the RSA method.

4.2 TESTING FOR DECRYPTION

The results of testing the second decryption technique are shown in Table 2 and figure 6. Blowfish's decryption procedure is faster than RSA's, just like its encryption. The speed percentage of Blowfish decryption against RSA is 165 percent, as shown in Table 2. RSA is 62.05 percent slower than Blowfish in all other ways.

Characters	Blowfish (ms)	RSA (ms)	Blowfish against RSA percentage (%)	RSA against Blowfish percentage (%)
10	2	5	150	60
50	4	10	150	60
100	6	18	200	66.66667
200	10	26	160	61.53846
Average			165	62.05

Table 2: Compares the time encryption of blowfish and RSA

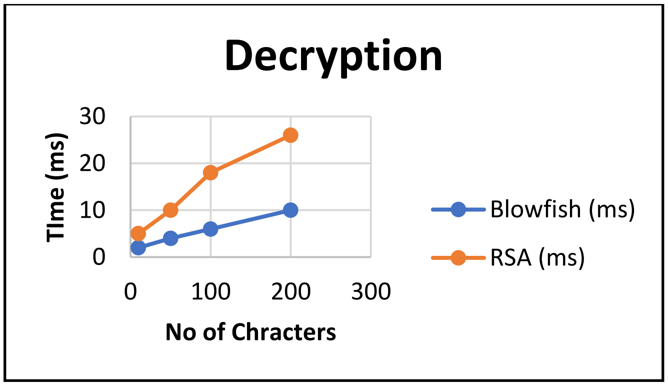


Fig 6: Graphical chart for Decryption

5. CONCLUSION

As data input environment in python, we have given the character inputs and based on that we have carried out certain conclusion. The outcome of this study is that the blowfish method is faster than RSA in terms of encryption or decryption. Encryption and decryption procedures will be used to encrypt and decrypt the quantity of characters. The blowfish encryption process is 237.5 percent faster than RSA, and the decryption process is 165.5 percent faster than RSA, according to testing data. RSA, on the other hand, performs 69.9 percent slower than blowfish in the encryption process. Furthermore, the decryption procedure is slower than blowfish, which has an efficiency of 62.5 percent.

6. REFERENCES:-

1. Asassfeh, Mahmoud & Qatawneh, Mohammad & Alazzeh, Feras. (2018). Performance Evaluation of Blowfish Algorithm on Supercomputer IMAN1. International journal of Computer Networks & Communications. 10. 43-53. 10.5121/ijcnc.2018.10205.
2. Nie, T., Song, C. and Zhi, X., 2010, April. Performance evaluation of DES and Blowfish algorithms. In Biomedical Engineering and Computer Science (ICBECS), 2010 International Conference on (pp. 1-4). IEEE.
3. Gatliff, B., 2003. Encrypting data with the Blowfish algorithm. Embedded Systems Programming, 16(8), pp.28-35.
4. Schneier, B., 1993, December. Description of a new variable-length key, 64-bit block cipher (Blowfish). In International Workshop on Fast Software Encryption (pp. 191-204). Springer, Berlin, Heidelberg.
5. Oukili, S. and Bri, S., 2016. High Throughput Parallel Implementation of Blowfish Algorithm. Applied Mathematics & Information Sciences, 10(6), pp.2087-2092.
6. H. Galli, —International Journal of Advanced Research in Computer Science and Software Engineering Data Security in Cloud using Hybrid Encryption and Decryption,‖ vol. 3, no. 10, pp. 494–497, 2013.
7. K. Aggarwal, —Performance Evaluation of RC6 , Blowfish , DES , IDEA , CAST-128 Block Ciphers,‖ vol. 68, no. 25, pp. 10–16, 2013.
8. Ramesh, Archana, and A. Suruliandi. "Performance analysis of encryption algorithms for Information Security." Circuits, Power and Computing Technologies (ICCPCT), 2013 International Conference on. IEEE, 2013.
9. D. Salama, A. Minaam, H. M. Abdual-kader, and M. M. Hadhoud, —Evaluating the Effects of Symmetric Cryptography Algorithms on Power Consumption for Different Data Types Evaluating the Effects of Symmetric Cryptography Algorithms on Power Consumption for Different Data Types,‖ no. October 2015, 2010.
10. M. Umaparvathi and D. K. Varughese, —Evaluation of Symmetric Encryption Algorithms for MANETs,‖ pp. 10–12, 2010.
11. B. L. Srinivas, A. Shanbhag, and A. S. D. Souza,—A Comparative Performance Analysis of DES and BLOWFISH Symmetric Algorithm,‖ pp. 77–88, 2014.
12. Thomas, Minta, and V. Panchami. "An encryption protocol for end-to-end secure transmission of SMS." Circuit, Power and Computing Technologies (ICCPCT), 2015 International Conference on. IEEE, 2015.

13. Himani Agrawal and Monisha Sharma ―Implementation and analysis of various symmetric cryptosystems‖ vol. 3, no. 12, pp. 1173–1176, 2010.
14. T. Nie, ―A Study of DES and Blowfish Encryption Algorithm,‖ pp. 1–4, 2009.
15. D. Suresh and M. L. Florence, ―RSA Algorithm & Signature In Cloud A Review Saas PaaS IaaS,‖ no. March, 2015.
16. P. V. P. Narkhede, M. P. S. Ajabe, S. M. Dandage, and M. P. B. Zope, ―A Review of Public Key Cryptography for Secure Communication Using RSA,‖ no. March, pp. 1–4, 2015.
17. Bokhari, M. U., Shallal Q. M., and Tamandani, Y. K. "A Performance Analysis of Hybrid Technique using DES and RSA algorithms," Computing for Sustainable Global Development (INDIACom), 2017 4th International Conference on IEEE, 2017.
18. Pugila, Dhananjay, et al. "An efficeient encrpytion algorithm based on public key cryptography." International Journal of Engineering and Technology 5.3 (2013): 3064-3067.
19. S. Rani, ―A Compound Algorithm Using Neural and AES for Encryption and Compare it with RSA and existing AES,‖ vol. 3, no. 1, pp. 35–38, 2015.
20. Bokhari, M. U., and Shallal, Q. M., "Evaluation of Hybrid Encryption Technique to Secure Data during Transmission in Cloud Computing," International Journal of Computer Applications, vol. 166, no. 4, 2017.
21. Bokhari, M. U., Shallal, Q. M., & Tamandani, Y. K. ―Reducing the Required Time and Power for Data Encryption and Decryption Using K-NN Machine Learning,‖ IETE Journal of Research, pp. 1- 9, 2018.