Using a Modified Approach of Blowfish Algorithm for Data Security in Cloud Computing

Reynaldo R. Corpuz
Technological Institute of the Philippines
938 Aurora Blvd., Cubao
Quezon City 1109 Philippines
reynaldo.r.corpuz@isu.edu.ph

Bobby D. Gerardo
West Visayas State University
Lapaz, Iloilo City
Philippines
bgerardo@wvsu.edu.ph

Ruji P. Medina
Technological Institute of the Philippines
938 Aurora Blvd., Cubao
Quezon City 1109 Philippines
ruji.medina@tip.edu.ph

ABSTRACT

Cloud computing brought remarkable potential changes and excellent opportunities to the information technology industry. It is particularly useful in most of the networked transactions. Cloud computing common issue problems associated with data security. Information is secure if it fulfills the three conditions namely Confidentiality, Integrity and Availability. In cloud computing confidentiality is obtained by cryptography. Symmetric algorithms notably Blowfish algorithm displayed undeniable success in cryptography. In this paper, a Modified Approach of Blowfish Algorithm using Shuffle Algorithm was applied and test the encryption, decryption speed, and throughput. The result shows that the higher the performance, the higher the efficiency of encrypting files.

CCS Concepts

• Security and privacy→Block and stream ciphers

Keywords

Cloud Computing; Blowfish Algorithm; S-box; permutation; Shuffle Algorithm; Encryption; Decryption.

1. INTRODUCTION

Cloud computing is a model for allowing pervasive computing, suitable for on-demand access to a network of reconstructing computing resources that can rapidly deliver and released with minimal effort or service provider interaction[1], [2]. Cloud computing brought remarkable potential changes and excellent opportunities to the information technology industry. It is particularly useful in most of the networked transactions. The most common problems associated with cloud computing are data privacy, security anonymity, authenticity, integrity, reliability and more [3]–[9].

Security becomes a big issue when a user stores essential information to the cloud server in which a user cannot directly control and maintain [10]. Data transmitted in the cloud storage is under threat because any unauthorized user can access and modify, there is a need the data to be secured all the time. Information is secure if it fulfills three conditions (1) Confidentiality, (2) Integrity, (3) Availability. Confidentiality meaning the data is being kept securely by not divulging to others; to prevent any unauthorized disclosure of the sensitive information. Integrity indicating the received data should be the same as the sender sends it; with this, it prevents modification from an unauthorized user. Availability refers to assurance that user has access to information anytime and to any network. In cloud computing confidentiality is obtained by cryptography.

A science of keeping message secure is called cryptography, that plays a vital role in information security against known attacks and decrease the risk of hacking information[11]. The unprecedented use of digital communications, electronic financial transactions, and digital transactions have brought remarkable potential on security issues and attacks against cryptography, hackers are undeniably much-sophisticated year-after-year[12]–[14]. There are types of cryptographic algorithms (1) symmetric algorithms (2) Asymmetric algorithms and (3) Hashing.

Hashing is a fixed length the signature created with the help of algorithms or hash function for the encryption of data. Each message consists of a different hash value, but the hashing has one drawback, i.e., when data is encrypted, the data cannot be decrypted. Symmetric and asymmetric algorithms removed this limitation of hashing. Symmetric algorithm is also known as "Secret Key Encryption Algorithm" in symmetric key algorithm, using only one key for encryption and decryption, i.e., private key, whereas in asymmetric algorithm both public and private keys are used for encryption and decryption, asymmetric algorithm is also known as "Public Key Encryption Algorithm[15].

Blowfish Algorithm has been widely analyzed and gradually accepted as a good and powerful cryptographic algorithm offering several advantages among which is its suitability and efficiency for implementing hardware and software; it satisfies the basic requirements in cryptography with high in attack immunity and relative low in algorithm complexity[16]–[18]. The algorithm is available to the public domain, unpatented and can be used by anyone for free. Blowfish (BF) algorithm displayed undeniable success in cryptography. However, several papers mentioned some of its drawbacks[16], [19]–[22].

These problems point out to the substitution and permutation of values in substitution box (S-box) which directly influenced in the overall encryption process. Improving the security of cryptosystem based on block cipher depends on the replacement of Substitution boxes in the encryption/decryption algorithm[23]. The generation of S-box depends on Pseudo-Random-Number-Generators and shared-secret-key. According to shared-secret-key all Pseudo-

ICIT 2018, December 29–31, 2018, Hong Kong, Hong Kong
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6629-8/18/12...$15.00
https://doi.org/10.1145/3301551.3301597

Random- Numbers are scrambled. After scrambling, the S-box is generated[24].

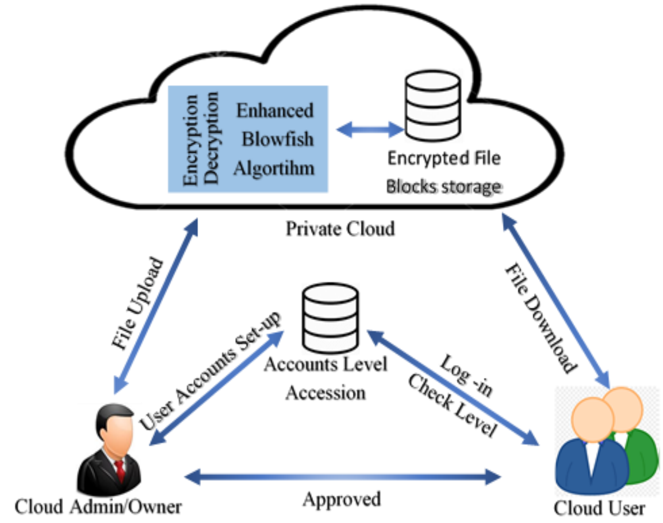


Figure 1. System Architecture

Figure 1 shows the system architecture of a private cloud includes Level of accession (LOA), cloud admin/owner (CAO), and cloud user (CU). CAO set-up the users account in the database to give access in the cloud; upload the data on the cloud server. The file is encrypted using the Modified approach of Blowfish Algorithm using Shuffle algorithm. Each file has its substitution box (S-box) and user key (input users key) stored in the private cloud server. Storing used Keys in the four S-box in random locations. Cloud computing is a multi-user environment. More than one user can access the file from the cloud server. CU request for a file by using the given LOA to log in and approved by the CAO to get the requested file. Each CU will be assigned an account level of accession. Level of Accession includes (a) ***0***- for the cloud admin/owner, enable to add/remove/edit/approve users account, and file upload/download; (b) ***1***- for the cloud user, enable to add/edit users account and upload/download file; (c) ***2***- for the cloud user, allow to download decrypted file.

This research work attempts to apply the Modified Approach of Blowfish Algorithm Based on S-box Permutation using Shuffle Algorithm in [11]. It specifically aims to Simulate and Test the Modified Blowfish Algorithm in Cloud Computing by using the encryption, decryption and throughput standards.

2. RELATED WORK

BF Algorithm is a symmetric block cipher. A 64-bit data block cipher and a variable-length secret-key range from any length to 448 bits[16]. The algorithm has fragments of two: for key-expansion and data encryption. The expansion key transforms a key of at most 448 bits into multiple subkey arrays with a total of 4168 bytes. Encryption of data occurs thru a 16 -round Feistel Network. For every round, it has a key-dependent permutation and a key-and-data-dependent switch. A 32-bit words additions and XORs for all operations Data lookups per round for the four indexed arrays are the only additional operations[25].

2.1 Expansion Keys

The BF algorithm uses a large number of subkeys. 18-P-arrays for 32-bit and 256 entries for each 32-bit S-boxes. All subkeys should perform before the encryption or decryption of data. Below is the exact process of BF:

1: With the use of hexadecimal digits of pi for the P-array initialization and then the four S-boxes

2: With the first 32-bit of the key will XOR to P1, then Second 32-bit of the key will XOR to P2 and so on for all the bits of the key until P18.

3: using the keys described in steps one and two all string zero must encrypt with the BA.

4: the output of step 3 replaces the values of P1 and P2.

5: using the Blowfish algorithm with the modified keys encrypt the output of step three.

6: output from step 5 replace P3 and P4.

7: the process continues, returning all the elements of the P-array, then all S-boxes one after the other, with the continuously changing output of the BA.

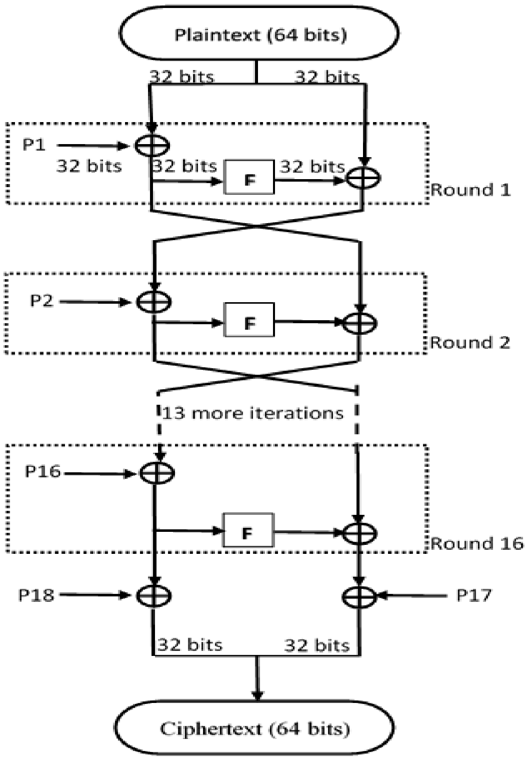


Figure 2. Block Diagram of Blowfish

2.2 Encryption of Data

As stated previously, as shown in Figure 2 blowfish algorithm is a Feistel network that consists of 16 rounds. It has an input of 64-bit plaintext and subkeys (32 bits) of 18 P-array. An output of 64 bits ciphertext. In algorithm 1 describing the Blowfish Algorithm.

```
**Input:** Plain text (64-bit) and from P1 to P18
**Output:** Ciphertext (64-bit)
Divide plain text into two, and each has 32-bit for XLeft and XRight;
**for** i=1 to 16 do
perform XLeft = XLeft XOR Pi;
```

perform XRight = F(XLeft) XOR XRight;

Swap XLeft and XRight (the last swap, undo.);

end

perform XRight = XRight XOR P17;

perform XLeft = XLeft XOR P18;

Recombine XLeft and XRight (ciphertext);

Algorithm 1: Blowfish Algorithm

Decryption is the same in the encryption process, except for P1-P8 are used in the reverse order.

Blowfish Algorithm has been exceptional in contrast to the other block cipher algorithms. In spite of its success, the S-Box permutation and F function need to enhance by improving its random permutation which significantly affected by the encryption process of the algorithm[26]. Modifying Blowfish Algorithm is conceivable by using Fisher-Yates shuffle algorithm (FYS) to enhance the random generation and permutation value of S-box and strengthening parallel execution of F function an additional key (user input key) added after performing the operators of S-boxes to further come up with more robust and satisfying result [11], [27], [28]. Figure 3 showing the modified structure of Function F.

2.3 A procedure of the S-box permutation

Table 2 showing the proposed substitution box using FYS algorithm

A.1. Initialize a linear array of SI of size 256 with values starting 0-256 in ascending order. Set Δ = length (SI)

A.2. Iterate the Piece-wise linear chaotic map for N0 times to die-out transient effect of the map with selected initial conditions.

A.3. Set counter= 1.

A.4. Further, iterate the map (1) and sample the chaotic state-variable x.

A.5. Extract a random number m ∈ [1, k] from x as:

$m = \{floor(x*1010)\}mod(k) +1$ where, k= Δ -counter+1

A.6. Exchange the two elements of array SI at positions m and k i.e SI(k) ↔ SI(m).

A.7. Set counter = counter + 1, if counter < 256 go to A.4.

A.8. re-apply the FYS on current SI by repeating steps A.3 to A.7 ξ times.

A.9. Translate resultant shuffled linear array SI to 16×16 table to get the final S-box.

2.4 The Modified Function F

In equation 2 of Function F showing modification without violating the security requirements of the BF and supports the parallel evaluation of 2-OR operations ($S_{1,\,a} + S_{4,\,d}$) mod 2^{32}) and ($S_{2,b} + S_{3,c}$) mod 2^{32}) and with the use of 2-XORs: one for the parallel operation and another XOR for the users' input key.

The users' input key (KI) further strengthens the S-box in the encryption process. Each 4-S-box values has KI randomly located within the S-box, to improve the quality of encryption[11].

$$F(XL) = (((S_{1,\,a} + S_{4,\,d})\ mod2^{32})\ XOR\ ((S_{2,\,b} + S_{3,c})\ mod2^{32})))XOR\ KI$$

S- Substitution box; **KI**- users Input key

Equation 2: Modified Function F

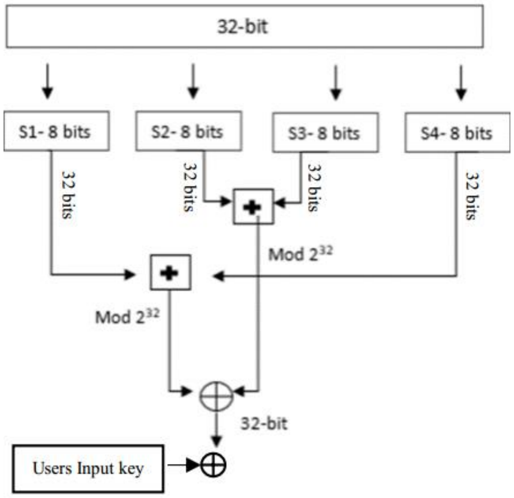


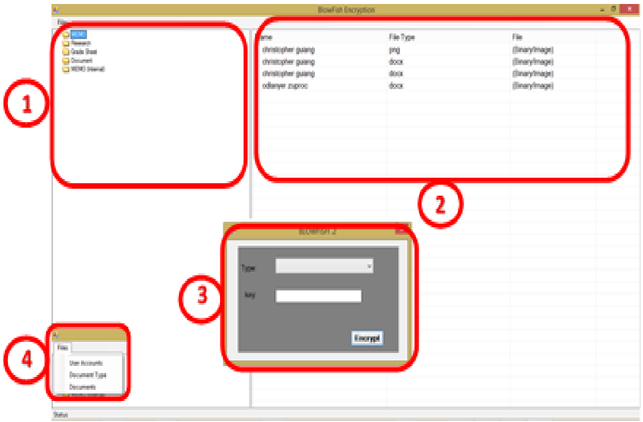
Figure 3. Modified Function F of Blowfish Algorithm

3. RESULTS AND DISCUSSIONS

The comparison and performance of the Modified Approach of Blowfish Algorithm with the Original Blowfish Algorithm. For evaluation, the following criteria are throughput, encryption time and decryption time. During the execution of both Algorithm was recorded using the different file formats like a word document, spreadsheet, picture and a portable document format (pdf).

Program interface and testing were done in the same platform using the C# programming language to obtain the fair comparison as shown in Figure 2. A simulation was performed using a Laptop computer with an AMD E2 1.70 GHz processing speed, 4GB RAM, and 1TB internal memory with 80% free disk space and a 64-bit operating system.

In Figure 4 showing the different parts of the program interface such as **(1)** *Folder panel*-showing all the files encrypted. **(2)** *Encrypted File Panel* allows the user to see all data according to the user's list. **(3)** *Encryption Window* it is where the user keyed its input key and assigning a specific folder for the encrypted data. **(4)***Drop Down List* showing the options setup like users account and document selection for encryption.



ame	File Type	File
christopher guang	png	(binaryImage)
christopher guang	docx	(binaryImage)
christopher guang	docx	(binaryImage)
odlanyer zuproc	docx	(binaryImage)

Figure 4. Program Interface

1. **Folder Panel** – allows the user to select a folder for the encrypted file

2. **Encrypted File Panel**- provides the list of coded data according to the users' list.
3. **Encryption window** – performs the encryption process
4. **Drop Down List** – provides list options to the user like account setup, document selection.

The file used was a real document taken from the Records Office of Isabela State University- Cauayan Campus for simulating the encryption, decryption process uses in this paper. The experimental runs were performed using the different file formats like word, spreadsheet, picture and pdf. Files have different contents.

3.1 Comparative Performance

The performance of the Modified BF and the Original BF was executed and operated for the different file formats and size to determine the processing speed. The result of the encryption and decryption and throughput are shown in figures 5- 9 respectively.

From the figures, 5-8 it can be seen that Modified BF consistently has the least processing time among the files used regardless of its content and size. In Figure 5 and Figure 6 showing the Encryption and Decryption Time using Different File Size processed in milliseconds, it denotes that in Modified Blowfish its encryption and decryption outperformed the Original Blowfish.

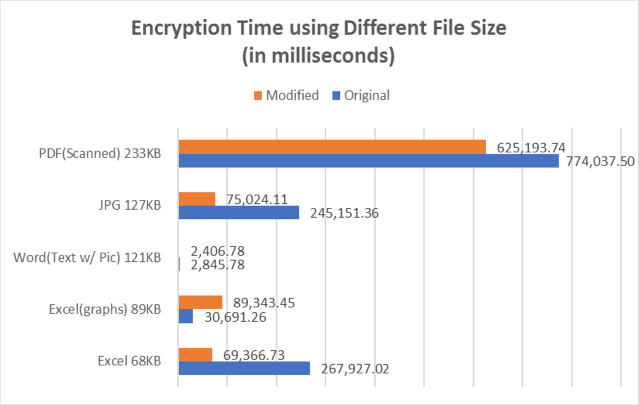


Figure 5. Encryption Time in Different File size

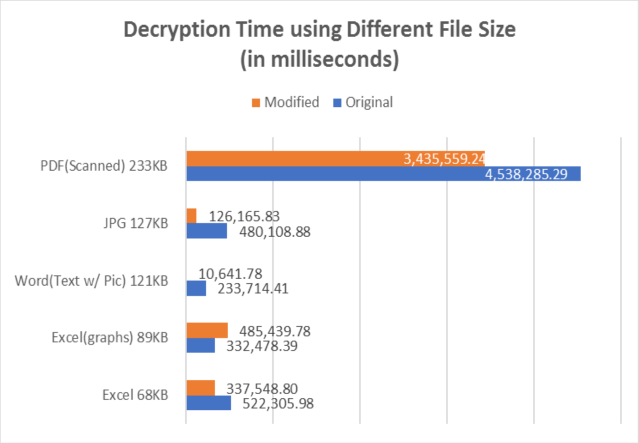


Figure 6. Decryption Time in Different File size

In Figure 7 and Figure 8 shows the efficiency of the Modified Blowfish Algorithm in encrypting and decrypting any file types. Figure 9 details the performance of both Modified and Original Blowfish as shown in the line graph that throughput for the Original Blowfish is lesser than the Modified Blowfish which means that the lower the performance, the less efficient, the higher the performance, the higher the efficiency of encrypting any files and size.

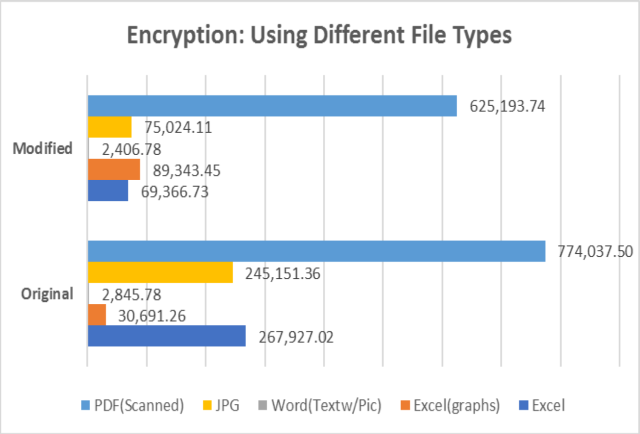


Figure 7. Encryption of Different File Types

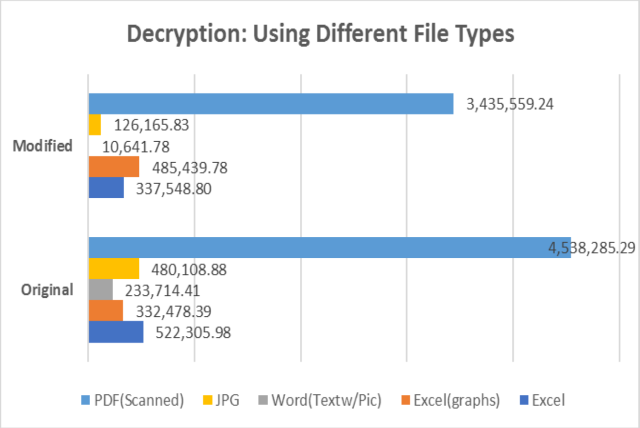


Figure 8. Decryption of Different File Types

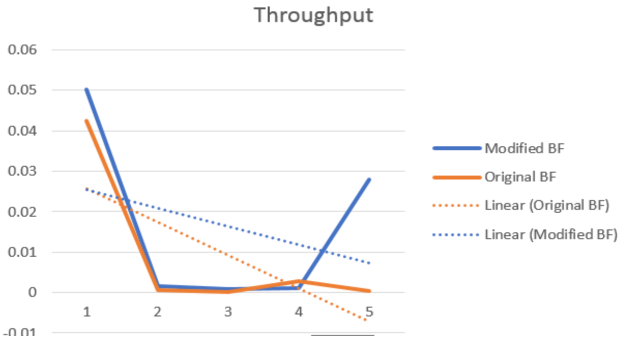
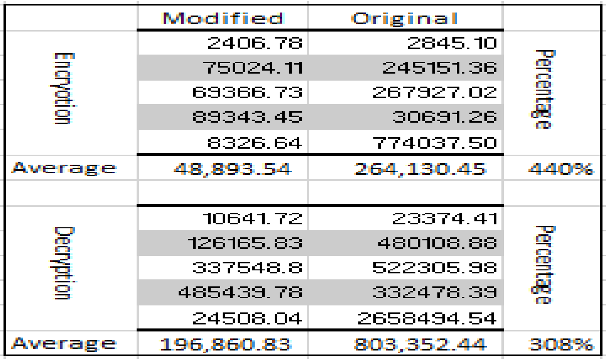


Figure 9. Processing Speed

The use of YFS and modified Function F improved the processing time of the Blowfish Algorithm. The improvement over the BF using the existing Function F and Random Permutation in S-Box showing in Table 1. From the table, it clearly shows that the performance of Modified BF is incomparable to that of the Original BF, revealing a massive difference in Encryption time with a percentage of 440% with that of the processing of all the files used encrypted. So as with the Decryption time of 308% respectively.

Table 1. Comparative improvement of BF using YFS and Modified Function F regarding encryption and Decryption time (in milliseconds)



	Modified	Original	
Encryption	2406.78	2845.10	Percentage
	75024.11	245151.36	
	69366.73	267927.02	
	89343.45	30691.26	
	8326.64	774037.50	
Average	48,893.54	264,130.45	440%
Decryption	10641.72	23374.41	Percentage
	126165.83	480108.88	
	337548.8	522305.98	
	485439.78	332478.39	
	24508.04	2658494.54	
Average	196,860.83	803,352.44	308%

4. CONCLUSION

This paper presented an Application of the Modified Approach of Blowfish Algorithm for Cloud computing in the Isabela State University using the Proposed System Architecture and User interface that add security to the File shared over Cloud computing. The simulation and testing were made possible using the developed interface. The file used was taken from the Records Office to obtain the results of this paper. The output of the application shows that the Modified approach of Blowfish for Cloud computing provides more efficient in encrypting and decrypting files. The test was also conducted to determine the quality and efficiency of the Modified approach of Blowfish. The result showed that the performance of the Modified Approach of Blowfish is efficient in encryption with a percentage of 440% for all the files and 308% for decryption respectively.

5. ACKNOWLEDGMENTS

Deepest gratitude to the Technological Institute of the Philippines (TIP), Commission on Higher Education (CHED) and Isabela State University (ISU). Also, to Dean of the Graduate Programs, Dr. Ruji P. Medina for his wisdom in keeping this paper a successful one and my Adviser, Dr. Bobby D. Gerardo.

6. REFERENCES

[1] B. Joshi, Karuna P, Theofanos, Mary, And Stanton, "Framework for Cloud Usability NIST Special Publication 500-316 Framework for Cloud Usability," pp. 1–18, 2015.

[2] H. Takabi, J. B. D. Joshi, and G. J. Ahn, "Security and privacy challenges in cloud computing environments," *IEEE Secur. Priv.*, vol. 8, no. 6, pp. 24–31, 2010.

[3] L. Ertaul, S. Singhal, and G. Saldamli, "Security Challenges in Cloud Computing," 2010.

[4] T. Ramaporkalai, "Security Algorithms in Cloud Computing SECURITY ISSUES OF CLOUD," vol. 5, no. 2, pp. 500–503, 2017.

[5] R. Ahmed and M. L. Ali, "Minimization of Security Issues in Cloud Computing," 2017.

[6] C. Paper and K. P. Siemens, "Cloud computing security issues and challenges," no. June 2010.

[7] R. Kaur and R. P. Singh, "Enhanced cloud computing security and integrity verification via novel encryption techniques," *Proc. 2014 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2014*, pp. 1227–1233, 2014.

[8] M. Shashi, "Cloud Computing Models : Background, Data security, & Security Issues," vol. 2, no. 2, pp. 1–6, 2017.

[9] H. Kaur, "A Novel Technique of Data Security in Cloud Computing based on Blowfish with the MD5 method," vol. 3, no. 6, pp. 828–837, 2017.

[10]A. Pansotra and S. P. Singh, "Cloud security algorithms," *Int. J. Secur. Its Appl.*, vol. 9, no. 10, pp. 353–360, 2015.

[11] R. R. Corpuz and B. D. Gerardo, "A Modified Approach of Blowfish Algorithm Based On S- Box Permutation using Shuffle Algorithm."

[12] M. Vanitha and R. Mangayarkarasi, "Comparative study of different cryptographic algorithms," *Int. J. Pharm. Technol.*, vol. 8, no. 4, pp. 26433–26438, 2016.

[13] B. Schneier, "Cryptographic design vulnerabilities," *Computer (Long. Beach. Calif).*, vol. 31, no. 9, pp. 29–33, 1998.

[14] S. Oukili and S. Bri, "High Throughput Parallel Implementation of Blowfish Algorithm," vol. 2092, no. 6, pp. 2087–2092, 2016.

[15] P. Patil, P. Narayankar, D. G. Narayan, and S. M. Meena, "A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA, and Blowfish," *Procedia Comput. Sci.*, vol. 78, no. December 2015, pp. 617–624, 2016.

[16] B. Schneier, "Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)," *Fast Softw. Encryption. Lect. Notes Comput. Sci. Cambridge Secur. Work. Proc. (December 1993)*, vol. 809, no. December 1993, pp. 191–204, 1994.

[17] A. Alabaichi, F. Ahmad, and R. Mahmod, "Security analysis of blowfish algorithm," *2013 2nd Int. Conf. Informatics Appl. ICIA 2013*, pp. 12–18, 2013.

[18] T. Nie and T. Zhang, "A study of DES and Blowfish encryption algorithm," *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, pp. 1–4, 2009.

[19] B. Schneier, "Academic: The Blowfish Encryption Algorithm—One Year Later - Schneier on Security," *Dr. Dobb's Journal, Sept. 1995.*, 1995.

[20] S. Vaudenay and E. N. S. Dmi, "On the Weak Keys of Blowfish."

[21] V. Rijmen, "Cryptanalysis and design of iterated block ciphers.... - Google Scholar," 1997.

[22] R. Zhang and L. Chen, "A block cipher using key-dependent S-box and P-boxes," *IEEE Int. Symp. Ind. Electron.*, pp. 1463–1468, 2008.

[23] P. Mroczkowski, "Generating Pseudorandom S-Boxes – a Method of Improving the Security of Cryptosystems Based on Block Ciphers," *J. Telecommun. Inf. Technol.*, vol. Nr 2, pp. 74–79, 2009.

[24] B. K. Maram and J. M. Gnanasekar, "Evaluation of Key-Dependent S-Box Based Data Security Algorithm using Hamming Distance and Balanced Output," *Journal*, vol. 5, no. 1, pp. 67–75, 2016.

[25] B. Schneier, “Description of a new Variable-Length Key, 64-Bit Block Cipher (Blowfish),” *Dr. Dobb’s Journal, April 1994*, 1994.

[26] H. E. H. D. H. Ahmed *et al.*, “A Novel Fast Image Encryption Scheme Based on 3D Chaotic Baker Maps,” *Int. J. Sci. Eng. Res.*, vol. 3, no. 1, pp. 150–154, 2013.

[27] M. Ahmad, P. M. Khan, and M. Z. Ansari, “A Simple and Efficient Key-Dependent S-Box Design Using Fisher-Yates Shuffle Technique,” *Commun. Comput. Inf. Sci.*, vol. 420 CCIS, pp. 540–550, 2014.

[28] M. Eberl, “The Fisher-Yates shuffle,” pp. 1–9, 2018.