Doc vs Internet

91.64% Originality	8.36% Similarity	58 Sources

Web sources: 58 sources found

1. https://achmatim.net/publikasi	1.94%
2. http://achmatim.net/publikasi	1.94%
3. https://docplayer.net/74948761-Optimization-of-lzw-compression-algorithm-with-modification-of-dic…	1.19%
4. https://jurnal.ugm.ac.id/ijccs/article/download/28707/19949	1.19%
5. https://journal.ugm.ac.id/ijccs/article/download/28707/19949	1.19%
6. https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-018-0591-0	0.72%
7. http://alumni.cs.ucr.edu/~yonghui/publications/dcc06.pdf	0.63%
8. https://link.springer.com/chapter/10.1007%2F978-981-10-5699-4_7	0.54%
9. http://paper.ijcsns.org/07_book/201602/20160215.pdf	0.54%
10. https://rd.springer.com/content/pdf/10.1007%2F978-981-10-1721-6_3.pdf	0.54%
11. http://www.freepatentsonline.com/9762907.html	0.51%
12. http://www.freepatentsonline.com/9769477.html	0.51%
13. https://ukm.pure.elsevier.com/en/publications/fast-dynamic-clustering-soap-messages-based-com…	0.48%
14. https://www.frontiersin.org/articles/10.3389/fict.2018.00002/full	0.48%
15. http://www.jurnal.unsyiah.ac.id/AICS-SciEng/article/download/1697/1596	0.42%
16. http://orion.lcg.ufrj.br/Dr.Dobbs/books/book3/chap10.htm	0.42%
17. https://symbiosisonlinepublishing.com/computer-science-technology/computerscience-informatio…	0.42%
18. https://www.sciencedirect.com/science/article/pii/S1877050916001101	0.39%
19. https://www.ijsce.org/download/volume-3-issue-5	0.39%
20. http://www.softpanorama.org/HPC/DNA_sequencing/index.shtml	0.36%
21. http://cs.ndsu.edu/~perrizo/saturday/VDMMbook.doc	0.36%
22. http://www.cs.ndsu.nodak.edu/%7Eperrizo/saturday/book/VDB_book.doc	0.36%
23. https://publishingrevolutions.wordpress.com/tag/digital	0.36%
24. http://thesai.org/Downloads/Volume9No6/Paper_25-Movement_Direction_Estimation_on_Video.pdf	0.33%
25. https://www.sec.gov/Archives/edgar/data/225211/000161577417001596/s105815_10k.htm	0.33%
26. http://www.citethisforme.com/topic-ideas/technology/Bibliography-51808230	0.33%
27. https://link.springer.com/chapter/10.1007/978-3-319-71928-3_35	0.33%
28. http://cv.uoc.edu/annotation/40250ff636d1e0502f55147cbb1a6121/603022/PID_00245395/PID_00..	0.33%
29. https://www.statista.com/statistics/778132/us-industrial-internet-of-things-impact-manufacturing-in…	0.33%
30. https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-7610.1976.tb00381.x	0.3%
31. https://www.slideshare.net/ijsrd/encryption-and-compression-of-audiovideo-data-using-enhanced-a..	0.3%
32. http://goddessenterprises.net/acronymdictionary.html	0.3%
33. https://knowledgehub.intelledox.com/v10/docs/web-service-soap	0.3%
34. https://stackoverflow.com/questions/33766750/soap-client-consuming-rest-web-services	0.3%
35. https://knowledgehub.intelledox.com/docs/web-service-soap	0.3%
36. https://www.usenix.org/events/imc05/tech/full_papers/aiello/aiello_html/imcdraft_onlycompressio…	0.27%
37. https://www.scilit.net/journals/545332/0/100	0.27%
38. https://journal.ugm.ac.id/ijccs/article/view/28707	0.27%
39. http://www.journaltocs.ac.uk/index.php?action=browse&subAction=pub&publisherID=3303&journa..	0.27%
40. https://en.wikipedia.org/wiki/LZ77_and_LZ78	0.27%
41. https://mattmahoney.net/dc/dce.html	0.27%
42. https://www.radioeng.cz/papers/2012-4.htm	0.27%
43. http://mattmahoney.net/dc/dce.html	0.27%
44. https://link.springer.com/article/10.1007%2Fs11276-016-1289-9	0.27%
45. https://jurnal.ugm.ac.id/ijccs/article/view/28707	0.27%
46. https://www.slideshare.net/dhavaldave1/cloud-computing-security-15453169	0.24%
47. https://patents.google.com/patent/US8553759B2/en	0.24%
48. https://patents.google.com/patent/US8717204B2/en	0.24%
49. https://www.e-education.psu.edu/geog862/print/l7.html	0.24%
50. http://david.carybros.com/html/data_compression.html	0.24%
51. http://eprints.undip.ac.id/view/year/2015.default.html	0.24%
52. https://www.bls.gov/opub/mlr/2018/article/employer-sponsored-health-insurance-premiums.htm	0.24%
53. http://archive.unu.edu/unupress/unupbooks/80473e/80473E0f.htm	0.24%
54. https://www.onlinelibrary.wiley.com/doi/abs/10.1002/asi.21416	0.24%
55. https://academic.oup.com/occmed/article/58/4/245/1481603	0.24%
56. https://quizlet.com/239272590/cis4361-information-assurance-and-security-chapter-2-analyzing-th..	0.24%
57. https://academic.oup.com/psychsocgerontology/article/55/3/S163/607081	0.24%
58. https://www.blaenkdenum.com/notes/java	0.24%

Similarity
Similarity from a chosen source
abc Possible character replacement
Citation
References

2018 International Seminar on Application for Technology of Information and Communication (iSemantic)

RESTful Web Service Optimization with Compression and Encryption Algorithm

Achmad Solichin
Faculty of Information Technology
Universitas Budi Luhur
Jakarta, Indonesia
achmad.solichin@budiluhur.ac.id

Mochammad Andika Putra
Informatics Department
Universitas Budi Luhur
Jakarta, Indonesia
mochammadandika9@gmail.com

Khaeri Diniari
Informatics Department
Universitas Budi Luhur
Jakarta, Indonesia
dinikhaeridiniari@gmail.com

Abstract—Along with the increasing number of mobile-based applications, the use of web services as a method of data exchange between systems also increased. With the web service technology, the exchange of data between systems with each other becomes easier even though it has a different platform. Increasing the amount of data requires more storage space on the server. Therefore, this research proposed the compression and encryption algorithm to optimize RESTful web service. We proposed LZW compression to reduce the storage space needed to store data on the server. We also proposed a Blowfish and AES-128 algorithm for securing data exchange through the internet network. The results show that all services made can run well and can be accessed through the Android application. In addition, the implementation of compression algorithm resulted in storage space savings of 38% with a compression rate of 1.78. The application of compression algorithm has an impact on the increase of computation time, although not significant.

Keywords—RESTful Web service, optimization, LZW compression, Blowfish encryption, AES-128 encryption

I. INTRODUCTION

Mobile device users in the world are increasing from time to time. Statista predicts the number of mobile device users in 2018 as much as 2.53 billon people or 36% of the total world population [1]. The StatCounter site also notes that in March 2018 51.43% of internet users accessed the site via mobile devices [2]. The growing number of mobile device users is driving the growth of the number of mobile-based applications and web-based data-service exchange services. With the web service technology, the exchange of data between systems with each other becomes easier even though it has a different platform.

Currently, the standard web-based data exchange service boils down to two standards: SOAP and REST [3]. SOAP stands for Simple Object Access Protocol. SOAP is a simple XML-based protocol to let applications exchange data or information over HTTP. The standard protocol HTTP makes it easier for the SOAP model to pass through firewalls and proxies without any modifications to the SOAP protocol. However, the lack of SOAP is the slowness of data transfer. Some recent research still uses SOAP-based web services but is optimized with compression algorithms to speed up data exchange [4]–[6]. They use compression algorithms like Tagged Sub-optimal Code (TSC), Huffman and J-bit encoding.

Meanwhile, REST stands for Representational State Transfer. REST is a set of principles for data exchange architecture through standard HTTP protocol. REST does not contain any additional messaging layers and focuses on design rules to create services without territory constraints. The client can access the resources using a unique URI, and the resource representation is returned. When accessing RESTful resources with the HTTP protocol, the resource URL acts as a resource identifier. The standard HTTP operations known by REST are GET, PUT, DELETE, POST, and HEAD. REST-based web service has several advantages over SOAP. For most servers, RESTful web services provide a good caching infrastructure over an HTTP GET method. It also offers an effective way of interacting with lightweight clients, such as smartphones.

Currently, REST-based web services are already implemented in various applications, especially on web services that have a major mobile device client [7]. In this research implemented RESTful web service in terms of data exchange.

Increasing the amount of data requires more storage space on the server. To save data storage on the server can be done in various ways. One way to save data storage on a server is to compress files or data before saving it on the server. One of the most popular and well-suited compression algorithms for text data compression is the LZW algorithm [8], [9]. In this research, we implement it to compress data before it saved in the database. This is because the LZW compression algorithm which is a dictionary-based compression is very effective in compressing text data that has many letters, words or repetitive sentences [10]. The more repertory of the combined strings in an additional dictionary on a plaintext the better the plaintext compression result [11]. The purpose of this implementation is to reduce the size of the storage space on the server database with the LZW compression algorithm.

The LZW algorithm forms a list of tables that will encode the sequence of symbols into the N-bit index contained in the table. Table size has 2^N dictionary list. The general principle of how the LZW algorithm works is to check each emerging character and then merge with the next character into a new string. If the

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

new string is not in the dictionary or has not been indexed then the new string will be added to the dictionary.

We also proposed encryption algorithm for securing data exchange through internet network. In addition to speed, the exchange of data through the Internet network also need to consider the safety factor. Currently, there are several commonly used encryption algorithms, including DES, 3DES, AES, Blowfish, RC4, and RSA. DES algorithm is an encryption algorithm that is quite old and has some security issues [12], [13]. The 3DES algorithm is a development of DES, but still lacks the time side of the slow process. Both algorithms are designed to be applied to hardware-based encryption. Meanwhile, the AES algorithm is a development of the DES that is set to standard by the US federal organizations. This algorithm is very secure because it uses keys 128, 192 or 256 bits.

Some studies have compared the encryption algorithm to determine the performance of each algorithm. It is quite difficult to determine the best encryption algorithm because of many factors that influence it. However, in this study, the encryption algorithm is determined based on two main criteria: speed and security. In terms of speed, most researchers who have compared the capabilities of some encryption algorithms agree to conclude that the Blowfish algorithm is the best [14]–[18]. Meanwhile, in terms of security, the best algorithm according to previous research is AES [12], [15], [19]. Therefore, this study used two algorithms in sequence: Blowfish and AES with 128-bit keys.

Blowfish encryption requires 64 bits of plaintext blocks as input and produces 64 bits of ciphertext. The key size for Blowfish can be selected in the range of 32 bits to 448 bits. The larger the bit size, the higher the level of security. The input blocks are split in half L0 and R0 where each contains 32 bits.

II. PROPOSED METHOD

A. The Application Architecture

Based on the above background, in this study, we developed a web-based data exchange model. We use the REST web service method because it has advantages in terms of speed and data security. Meanwhile, to optimize data security in web service data exchange process, we utilize the REST web service with the Blowfish and AES-128 encryption algorithm. And to improve the effectiveness of data storage on the server, we apply the LZW compression algorithm. In this study also developed a mobile application that implements the proposed model.

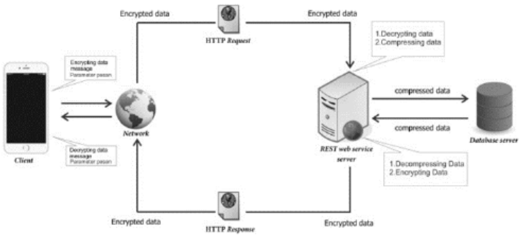

Fig. 1. The application architecture of the Proposed Model

TABLE I. WEB SERVICE DESIGN

Services	Method	Parameter	Response
Class list	GET		Class data (array)
Join Class	POST	id_kelas (integer)	Status (bool), message (string)
Unjoin Class	POST	id_kelas (integer)	Status (bool), message (string)
Class search	POST	carikelas (string)	Class data (array)
Class Delete	POST	id_kelas (integer)	Status (bool), message (string)
Create class	POST	Class data (array)	Status (bool), message (string)
Update class	POST	id_kelas (integer), Class data (array)	Status (bool), message (string)
User presence	POST	id_kelas (integer), id_user (integer)	Status (bool), message (string)
User Profile	GET	id_user (integer)	User data (array)
User followers List	GET	-	User follower data (array)
User following list	GET	-	User following data (array)
Add user skill	POST	id_user (integer), Skill data (array)	Status (bool), message (string)
Find user based-on event	POST	Keyword (string), id_event (integer)	User data (array)
Find user based-on name	POST	Keyword (string), id_event (integer)	User data (array)
Find user based-on skill	POST	Keyword (string), id_event (integer)	User data (array)
Follow an user	POST	id_user (integer), id_user2 (integer)	Status (bool), message (string)
Unfollow user	POST	id_user (integer), id_user2 (integer)	Status (bool), message (string)

Figure 1 presents the data exchange model proposed in this study. When a client requests a service, the data is encrypted using Blowfish and produce cyphertext of original data. Then the resulting ciphertext is encrypted again with the AES-128 algorithm to produce the final ciphertext. Upon receipt of the request, the web service decrypts the data and executes commands on request from the client. Similarly, when the web service sends the response data to the client, the data is encrypted first. The implementation of the encryption algorithm makes the exchange of data between the client and the web server to be more secure.

Meanwhile, the LZW compression algorithm is also applied when the application stores data to the database server. Before it is stored, the data is compressed in advance so as to save the storage capacity in the database server. In contrast, when retrieving data from a database, it is decompressed to restore the data to its original format. This data compression will save storage space on the server.

B. Web service design

To apply the optimized web service model with Blowfish + AES-128 encryption algorithm and the LZW compression, we proposed a REST-based web service that serves multiple requests. The service request method consists of a POST and GET. Table 1 lists the services provided by the web service.

III. RESULTS AND DISCUSSIONS

This section describes the prototype of mobile-based applications that implement encryption and compression algorithms, test results against applications and discussions.

A. The Prototype of the Proposed Method

In this study, we developed a mobile-based application that implements the Blowfish and AES-128 encryption algorithms, as well as the LZW compression algorithm. The developed application is part of BluCampus application of Budi Luhur University. The application displays services and information related classes in the form of courses, workshops, and seminars that can be followed by students of Budi Luhur University. Each user can register and follow the ongoing activities. The activity committee can also record the attendance of the participants through its mobile device.

Figure 2 presents some of the mobile-based application views developed in this study. Figure 2 (a) shows the list of active and participatory classes. Meanwhile, in Figure 2 (b) a page to create a class or course is shown, and Figure 2 (c) displays the feature of recording attendance in a class or activity.

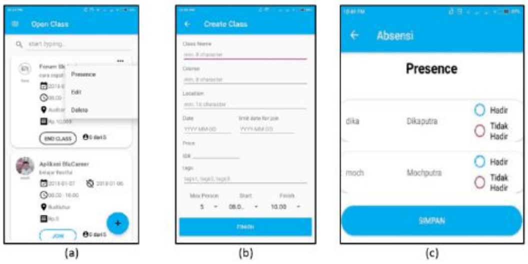

Fig. 2. Some of the application views, (a) active class list, (b) class creation form, and (c) attendance recording feature.

B. Encryption Testing on Data Exchange

Blowfish and AES-128 encryption algorithms are applied to the process of sending data from client to server and vice versa. Figure 3 shows an example of sending unencrypted data and Figure 4 shows an example of sending encrypted data. With the application of encryption algorithm in the data delivery process, data security will be better than sending data without encrypted.

C. Compression Testing

In Table 2 we present the results of compression and decompression process testing on a number of data. Test data in the form of text with a character length of 100, 200, 400, 800 and 1600. Each data is tested 10 times using the device RESTful Stress Tool and internet speed of 10 Mbps. The compression ratio (CR) and the Saving ratio (SR) are calculated with (1) and (2) [20], [21].

$$CR = \frac{uncompressed\blacklozenge size}{compressed\blacklozenge size} \quad (1)$$

$$SR = 1 - \frac{compressed\blacklozenge size}{uncompressed\blacklozenge size} \quad (2)$$

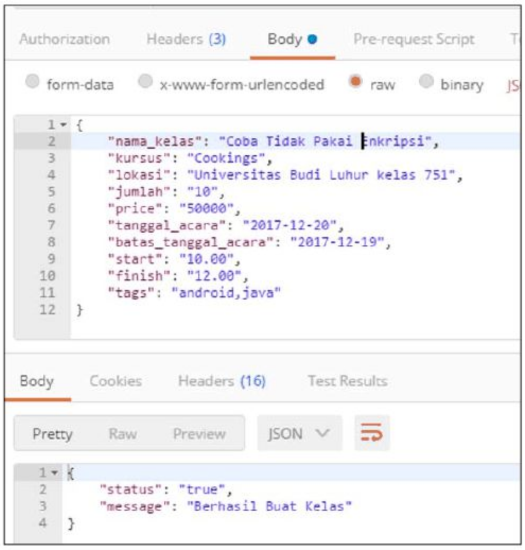

Fig. 3. Unencrypted data exchange

Fig. 4. Encrypted data exchange

TABLE II. THE EXPERIMENT RESULTS OF THE COMPRESSION ALGORITHM

Charac-ters	Original size (bytes)	Compressed size (bytes)	Comp. times (ms)	CR	SR	Dcomp. times (ms)
100	102	86	153	1.19	0.16	137
200	202	145	160	1.39	0.28	140
400	402	259	192	1.55	0.36	154
800	802	461	218	1.74	0.43	166
1600	1602	783	260	3.01	0.67	206
			Average	1.78	0.38	

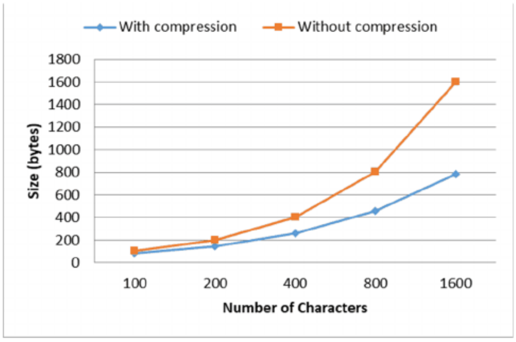

Fig. 5. Comparison of compressed file sizes with uncompressed files.

Test results showed a significant change in the size of the data after compression. The average compression rate of all tests was 1.78, and the resulting storage capacity savings were 38%. From the test results can also be concluded that the bigger the file size, the higher the compression ratio and the more storage space savings. Figure 5 presents a comparison of the size of the original data with the compressed data. In the meantime, Figure 6 shows the time comparison to store data between compressed and uncompressed data for every testing data.

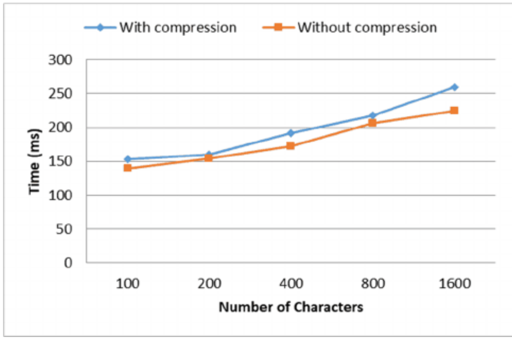

Fig. 6. Time comparison to store data between compressed and uncompressed data.

D. Web Service Testing

In this research, we also tested the web service with Blackbox testing method. Table 3 presents some of the web service services tested in this study. Based on the test results can be concluded that all services can run well.

IV. CONCLUSION

Based on the results of tests that have been done in this study, can be drawn some conclusions. This research produced a web service with several services. In addition, this study also produced a prototype of mobile-based applications. In testing the process of encryption of the data, all tested data successfully exchanged between web service and client. Web service testing showed that all services could run well.

TABLE III. THE TESTING RESULTS OF THE WEB SERVICE EXPERIMENTS

Num	Services	Results
1	List of the class	Success
2	Create new class	Success
3	Details of the class	Success
4	Update class	Success
5	Delete class	Success
6	Find class	Success
7	Join class	Success
8	Unjoin class	Success
9	List of joined class	Success
10	Class presence	Success
11	End the class	Success
12	User profile data	Success
13	User feedback data	Success
14	Create feedback	Success
15	User skill data	Success
16	Another user skill user	Success
17	Add new skill user	Success
18	Add user about me	Success
19	Delete skill	Success
20	List of the class of a user	Success
21	Class rating of the user.	Success
22	Class data from another user	Success
23	Class data of the joined users	Success
24	Class data of the another joined-user	Success
25	Seminar data of the joined user	Success
26	Seminar data of the another joined-user	Success
27	Workshop data of the joined user	Success
28	Workshop data of the another joined-user	Success
29	Find user data based on event	Success
30	Find user data based on the name	Success
31	Find user data based on skill	Success
32	List of the user followers	Success
33	List of the user following	Success
34	List of the followers of another user.	Success
35	List of the following of another user.	Success
36	Follow a user	Success
37	Unfollow a user	Success
38	List of user conversations	Success
39	The message of the user data	Success
40	Create message	Success

In the test of the compression algorithm, it is concluded that the average compression ratio of 1.78. By compressing the data, the average savings of storage space on the server as much as 38%. The application of compression algorithm has an impact on the increase of computation time, although not significant. It shows that the compression algorithm can optimize the implementation of web service.

REFERENCES

[1] Statista, "Number of smartphone users worldwide from 2014 to 2020 (in billions)," Statista.Com, 2018. [Online]. Available: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/. [Accessed: 01-Apr-2018].

[2] StatCounter, "Desktop vs Mobile vs Tablet Market Share Worldwide," StatCounter.Com, 2018. [Online]. bleAvaila http://gs.statcounter.com/platform-market-share/desktop-mobile-tablet. [Accessed: 02-Apr-2018].

[3] M. Zur Muehlen, J. V. Nickerson, and K. D. Swenson, "Developing web services choreography standards - The case of REST vs. SOAP," Decis. Support Syst., vol. 40, no. 1 SPEC. ISS., pp. 9–29, 2005.

[4] I. Al Rassan and H. Alyahya, "Tagged-sub Optimal Code (TSC) compression for improving performance of web services," Procedia

Comput. Sci., vol. 62, no. December 2015, pp. 167–169, 2015.

[5] A. R. Shukla and P. P. D. Thakare, "A Review On Performance Enhancement Of Web Services Using Tagged-Sub Optimal Code (TSC) Compression," Int. Res. J. Eng. Technol., vol. 5, no. 3, pp. 1177–1180, 2018.

[6] A. M. Abbas, A. A. Bakar, and M. Z. Ahmad, "Fast dynamic clustering SOAP messages based compression and aggregation model for enhanced performance of Web services," J. Netw. Comput. Appl., vol. 41, no. 1, pp. 80–88, 2014.

[7] H. Hamad, M. Saad, and R. Abed, "Performance Evaluation of RESTful Web Services for Mobile Devices.," Int. Arab J. e-Technol., vol. 1, no. 3, pp. 72–78, 2010.

[8] T. A. Welch, "A Technique for High-Performance Data Compression," IEEE Comput., pp. 8–19, 1984.

[9] J. Ziv and A. Lempel, "Compression of Individual Sequences via Variable-Rate Coding," IEEE Trans. Inf. Theory, vol. 24, no. 5, pp. 530–536, 1978.

[10] R. Maulunida and A. Solichin, "Optimization of LZW Compression Algorithm With Modification of Dictionary Formation," IJCCS (Indonesian J. Comput. Cybern. Syst., vol. 12, no. 1, p. 73, 2018.

[11] Hidayat, W. Zarman, and T. Pamungkas, "Implementasi Algoritma Kompresi Lzw Pada Database Server," J. Ilm. Komput. dan Inform. (KOMPUTA), vol. 2, no. 1, pp. 7–13, 2013.

[12] T. Pornin, "Comparison of DES, Triple DES, AES, blowfish encryption for data," StackOverflow.Com, 2011. [Online]. Available: https://stackoverflow.com/questions/5554526/comparison-of-des-triple-des-aes-blowfish-encryption-for-data. [Accessed: 24-Apr-2018].

[13] A. Solichin and E. W. Ramadhan, "Enhancing data security using DES-based cryptography and DCT-based steganography," in 2017 3rd International Conference on Science in Information Technology (ICSITech), 2017, pp. 618–621.

[14] J. Thakur and N. Kumar, "DES, AES and Blowfish: Symmetric key cryptography algorithms simulation based performance analysis," Int. J. Emerg. Technol. Adv. Eng., vol. 1, no. 2, pp. 6–12, 2011.

[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] S. P. Singh and R. Maini, "Comparison of Data Encryption Algorithms," Int. J. Comput. Sci. Commun., vol. 2, no. 1, pp. 125–127, 2011.

[17] A. Nadeem and M. Y. Javed, "A Performance Comparison of Data Encryption Algorithms," in 2005 International Conference on Information and Communication Technologies, 2005, pp. 84–89.

[18] D. S. A. Elminaam, H. M. A. Kader, and M. M. Hadhoud, "Performance Evaluation of Symmetric Encryption Algorithms," IJCSNS Int. J. Comput. Sci. Netw. Secur., vol. 8, no. 12, pp. 280–286, 2008.

[19] P. Princy, "A Comparison of Symmetric Key Algorithms DES , AES , Blowfish, RC4, RC6: A Survey," Int. J. Comput. Sci. Eng. Technol., vol. 6, no. 5, pp. 328–331, 2015.

[20] Wikipedia, "Data compression ratio," Wikipedia.com, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Data_compression_ratio#cite_note-CompressionRatioBroadcastEngineering-1. [Accessed: 10-Mar-2018].

[21] C. Poynton, Digital Video and HD: Algorithms and Interface, Second Edi., no. November. Elsevier Inc., 2012.