International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

Performance Analysis of Data Encryption Algorithm

Prathamesh P. Churi

Abstract*: *In this paper, author have performed experimental analysis of Jumbling-Salting (JS) algorithm for larger text size. In the earlier research work, JS Algorithm was symmetric-password encryption algorithm. JS algorithm consists of two prominent cryptographic processes namely Jumbling and Salting. Jumbling consists of three major randomized processes viz. Addition, Selection and Reverse. Jumbling process jumbles the random characters into password string. Salting process adds a random string based on some timestamp value. The output of Jumbling and Salting process is given to predefined AES block to perform 128-bit key encryption to maintain the cipher text size uniform. In this research, the capability of JS algorithm is enhanced. The paper therefore shows the performance of JS algorithm regarding length of cipher text size with respect to AES and DES algorithms. This extended research ensures that JS algorithm is not only suitable for smaller text like password, pin, passcode etc. but it is also favorable for larger texts.

Keywords: Jumbling, Salting, JS algorithm, Cloud, plain text, cipher text, AES, DES.

I. INTRODUCTION

According to Bruce Schneider et al. [1, 2, 3] "Security is a process, not a product." The deep meaning of this quote is based on a phenomenon that there may be many security techniques and methodologies available in a market, but none of them can be fruitful and guarantees that the security is 100% success.

We must adhere to the fact that the process which we follow in security must address and satisfy all the security goals [4] in an organization. Authentication is the common phenomenon in security which grants/deny the permission to access the system resources by means of unique key/passcode /pattern. [5] The password is one of such common authentication technique which provides the claimant access to system resources. Being the simplest form of authentication technique used, the probability of cracking the password using different combinations is considerably high [2, 3]. The possible common attacks possible to obtain a password is - by attacking the server's database which consists of a list of passwords. There exists brute force /dictionary attack which has proven this statement to be inefficacious. To overcome the problem of securing encrypted password, the author of this paper have developed JS (Jumbling –Salting) technique [2, 3] which will increase the length of cipher text by jumbling additional characters to the original set of a password.

JS algorithm majorly has two processes namely jumbling and salting [2, 3]. Jumbling process involves the prepending characters and jumbling them by use of mathematical modulus function. This can be implemented by three sub processes namely - "addition", "selection" and "reverse".

After the jumbling process the salting process works on jumbled string and adds the salt string which is a unique timestamp value obtained from server [2, 3]. jumbled and salted password is given to the predefined AES algorithm procedure to maintain the uniformity of the string.

The AES can be chosen as it is advanced encryption process for any data encryption technique with the limitations of sharing the key in the network and the length of cipher text output [2, 3]

The author had compared the result of above algorithm with the help of four analysis parameters [2, 3] namely,

1. Size of cipher text,
2. Encryption time,
3. Decryption time and
4. Throughput

It is observed that the parameter 1 - Size of cipher text has got excellent result. the size of password which is formed after encryption process is much larger and strong as compared to other symmetric encryption key algorithms like AES and DES [2, 3] The major aim was to encrypt strong string of password which can be susceptible to brute force/ dictionary attacks.

In order to extend the work, author have tried to apply the same algorithm in case of larger text of having 10000 characters, may be a file. It is observed that JS algorithm gives susceptible results which is much better to AES and DES algorithm.

The major aim of author is to apply the JS algorithm to the files which are stored to cloud [6]. We have used only one analysis parameter (Size of cipher text) for JS algorithm but for the larger text string. The initial size of the plain text is kept at 10000 and later on, it reaches to 10000000. The corresponding plain text and cipher text is measured. The encryption time is also measured. The author have got fruitful results for their experiment.

To continue the work, author are trying how JS algorithm will work in future for Cloud file encryption. The existing cloud encryption used AES algorithm as a major encryption algorithm. AES algorithm has a limitation of sharing key over a network [4] to augment this fact,

Manuscript published on 30 September 2019
* Correspondence Author
Prathamesh Churi*, Department of Computer Engineering, Mukesh Patel School of Technology Management and Engineering, NMIMS University, Mumbai, India. Email: Prathamesh.churi@nmims.edu / prathamesh.churi@gmail.com

author ensures that JS algorithm will give a fruitful response in case of file encryption. It is always said that randomness is ubiquitous in cryptography. More randomly you design an algorithm, more difficult to crack it. The processes involved in JS algorithm are purely randomized; hence we can achieve "Randomness in Security [7]

II. LITERATURE SURVEY

The table I illustrated below gives the methodologies/ Algorithms which are used for file encryption in cloud: (Highlighted points shows the limitations of algorithms.)

Table- I: Name of the Table that justify the values

Sr.no	Algorithm	Base algorithm	File type	Description
1	Hybrid Encryption Algorithm I	AES and RSA	Text	This paper mainly focuses on 1. Secure Upload of data on cloud so that even the administrator is unaware of the contents. **2. Download of data mainitaing integrity** 3. Proper usage and sharing of the public, private and secret keys involved for encryption and decryption.[8]
2	Hybrid Encryption Algorithm II	AES/3DES , RC6 , BRA	Text and Images	1. In this algorithm AES, blowfish, RC6 and BRA algorithms are used to provide block wise security to data. **2. For all cases , algorithm key size is 128 bit.** 3. LSB steganography technique is introduced for key information security. [9]
3	Transparent File Encryption	AES	Text	1. In the algorithm, a transparent file encryption system based on File System in User Space is proposed to overcome the shortcomings of the traditional transparent file encryption systems. 2. To avoid cleaning the system file cache frequently, file redirection is adopted to transfer file operations to FUSE.[10]
4	Secure Access Controlled File Encryption (SAFE) System	AES	Text	1. The design of Secure Access controlled File Encryption system to achieve policy-based access control and assured deletion. [11]
5	Attribute based Encryption	CP-ABE-WP Scheme	Text, Hierarchical Files	1.The authors give a systematic definition of attribute computing in cloud computing environment. 2. Based on the definition, we design a secure and practical attribute based encryption scheme without pairings (CP-ABE-WP) under cloud computing scenarios. [12,13]
6	AES-CTR based Algorithm	AES (CTR Mode – SHA256 Hash key generation)	Text files of larger size	1. Method allows efficient updates of encrypted files by minimizing the amount of data that need to be re-encrypted. 2. Achieves significantly better performance than full re-encryption for file updates [14]. **3. Only applicable for text data.**
7	Lattice based encryption algorithm	AES + R-LWE	Text, Image, PDF ,Excel ,Video	1. Implemented a lattice-based encryption algorithm based on hardness of Ring LWE problem, which is secure against quantum computing to secure files stored in Cloud Storage. [15] **2. Not applicable for larger files.**
8	Spark Encryption Algorithm	AES	Video	1. Use of AES Encryption for **specific slice of video** [16] **2. Not applicable for all the types of video**
9	Image Sharing Encryption Algorithm	AES 128 (CTR Mode)	Image	1. Use of AES Encryption for JPEG Image [17] **2. Lossy algorithm**
10	A Novel Image Encryption Method	EET , Complex Conjugate and Random Phase Key	Image	1. The authors proposed a new technique to encrypt an image for secure image transmission and parallel decryption using cloud resources. 2. The encoding of the image is done using the FFT conjugate complex property and private random phase key value shared between sender and receiver [18].

Table I, illustrates following inferences:

The algorithms studied from literature, are specific to file type. If the algorithms are used for some other files apart from what it is desired for, then it gives anonymous results.

- All algorithms use AES 128, 256 bit algorithms. Sharing and compromising key is the major issue in AES algorithm.
- Some of the algorithms has highest computational complexity.

III. JS ALGORITHM AND METHODOLOGY

A. System Block Diagram

The system block diagram of existing JS algorithm is shown in figure. 1

Jumbling block:

Jumbling block majorly dominated by three sub processes viz. Addition, Selection,

and Reverse process. The plain text password is given to Process array of this block [2, 3]. Jumbling block randomly adds some characters from character set and jumbling them with the help mathematical modulus function which is a mathematical function which gives the remainder of a given operation.

Addition sub-block of Jumbling process:

The entire algorithm works by generating principle random value [2, 3]. We accept the plain text password in an array. The length of an array is extended by the random number generated.

Selection sub-block of Jumbling process:

The empty blocks of an array are filled by different random characters from predefined character set. Randomly adding characters will be decided based on programmer-defined mathematical function. In General, there will be N number of character set. Characters in the character set may be repeated based on programmer's logic.

Reverse sub-block of Jumbling process:

The reverse function is called based upon predefined mathematical function or condition.

Salting block:

Salting is the process of adding random string of timestamp from confusion array. The timestamp value is the server's timestamp value.

The output of the previous block of array is then given to AES 256 in order to make the output of uniform length.

AES block:

Predefined 128 bit AES algorithm is applied to Jumbled-salted text password.

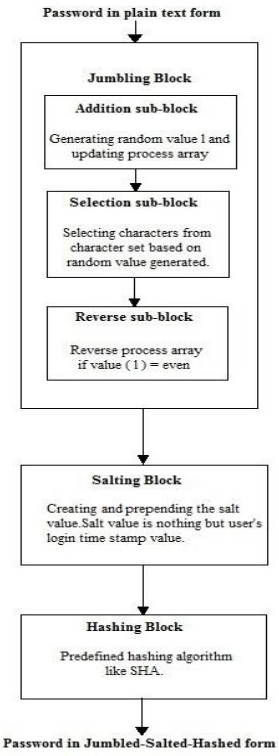

Fig. 1.Block Diagram of Jumbling-Salting Algorithm

B. Size of Cipher Text of plain text passwords

Table II shows the performance analysis of JS algorithm in case of password encryption. size of cipher text in JS, AES and DES algorithms.

Table II: Performance analysis of plain text password

TEXT	Plain text size(in bytes)	JS algorithm	AES algorithm	DES algorithm
		Decryption text size (in bytes)		
1	20	128	48	48
2	22	128	48	48
3	28	206	48	192
4	138	400	214	310
5	570	1360	836	1066

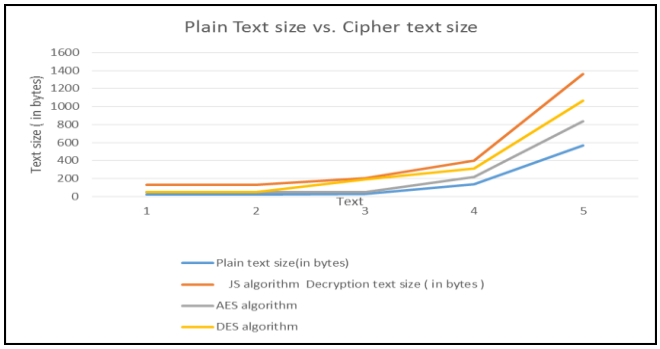

Fig. 2.Cipher text comparison between JS, AES and DES algorithms

Fig 2 shows the statistical analysis of Plain text size vs. Cipher text size of JS, AES, and DES algorithm for password encryption. Following inference can be made for following results.

- The size of cipher text of JS algorithm as compared to AES and DES algorithms is higher. There is exponential increase of cipher text of JS algorithm as compared to AES and DES algorithm
- The maximum number of random characters that can be added in jumbling block will range from X to 5 times the value of X. In future use it can be increased depends on programmer's logic.
- Additionally, the salt has been added in the salting process which increases the size cipher text drastically.

The results of this algorithm motivated us to perform encryption process in case of larger texts. The paper will show the statistical analysis of JS algorithm in case of larger text of million bytes of text data set. The inferences will be noted down in the results section of the same paper.

IV. EXPERIMENTAL ANALYSIS OF JS ALGORITHM FOR LARGER TEXT

The experiment is performed on following conditions. The care was taken that no background process is running on host machine so that there will be fairness in evaluation of Size of Cipher text vs. Plain text.

- Operating system: Windows 10, 64 bit operating system, Home Edition
- Processor: Intel (R) Core (TM) i5-5200U
-

- RAM: 4 GB
- CPU speed: 2.20 GHz -
- Programming Language: Visual Studio (version 2015)

A. Length of the Cipher text for larger text files

The Experiment is performed on 500 different text files for encryption. The result of first 30 readings are noted down. The cipher text length and corresponding encryption time is noted.

Table III. Performance of JS algorithm in larger text size

Data Set	Size of Plain Text (in bytes)	Size of Cipher Text (in bytes)			Encryption Time of JS algorithm
		JS	AES	DES	
1	200	**412**	**232**	**226**	412
2	400	**904**	**464**	**452**	424
3	800	**1864**	**928**	**904**	441
4	1600	**3738**	**1856**	**1808**	461
5	3200	**8563**	**3712**	**3616**	501
6	6400	**16438**	**7424**	**7232**	556
7	12800	**35478**	**14848**	**14464**	606
8	25600	**71350**	**29696**	**28928**	618
9	51200	**142756**	**59392**	**57856**	639
10	102400	**285512**	**118784**	**115712**	698
11	204800	**571024**	**237568**	**231424**	731
12	409600	**1142048**	**475136**	**462848**	779
13	819200	**2284096**	**950272**	**925696**	792
14	1638400	**4568192**	**1900544**	**1851392**	807
15	3276800	**9136384**	**3801088**	**3702784**	828
16	6553600	**18272768**	**7602176**	**7405568**	842
17	13107200	**36545536**	**15204352**	**14811136**	886
18	26214400	**73091072**	**30408704**	**29622272**	937
19	52428800	**146182144**	**60817408**	**59244544**	969
20	104857600	**292364288**	**121634816**	**118489088**	1000
21	209715200	**584728576**	**243269632**	**236978176**	1023
22	419430400	**1169457152**	**486539264**	**473956352**	1059
23	838860800	**2338914304**	**973078528**	**947912704**	1118
24	1677721600	**4677828608**	**1946157056**	**1895825408**	1190
25	3355443200	**9355657216**	**3892314112**	**3791650816**	1202

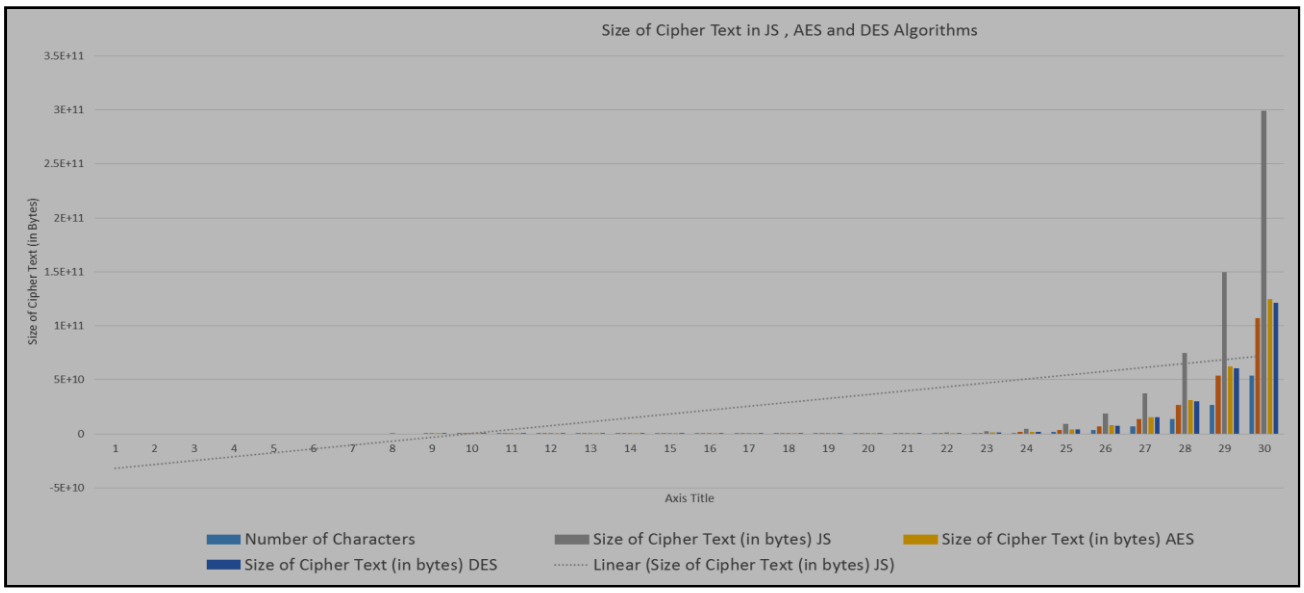

Fig. 3.Size of Cipher text in JS, AES and DES Algorithms

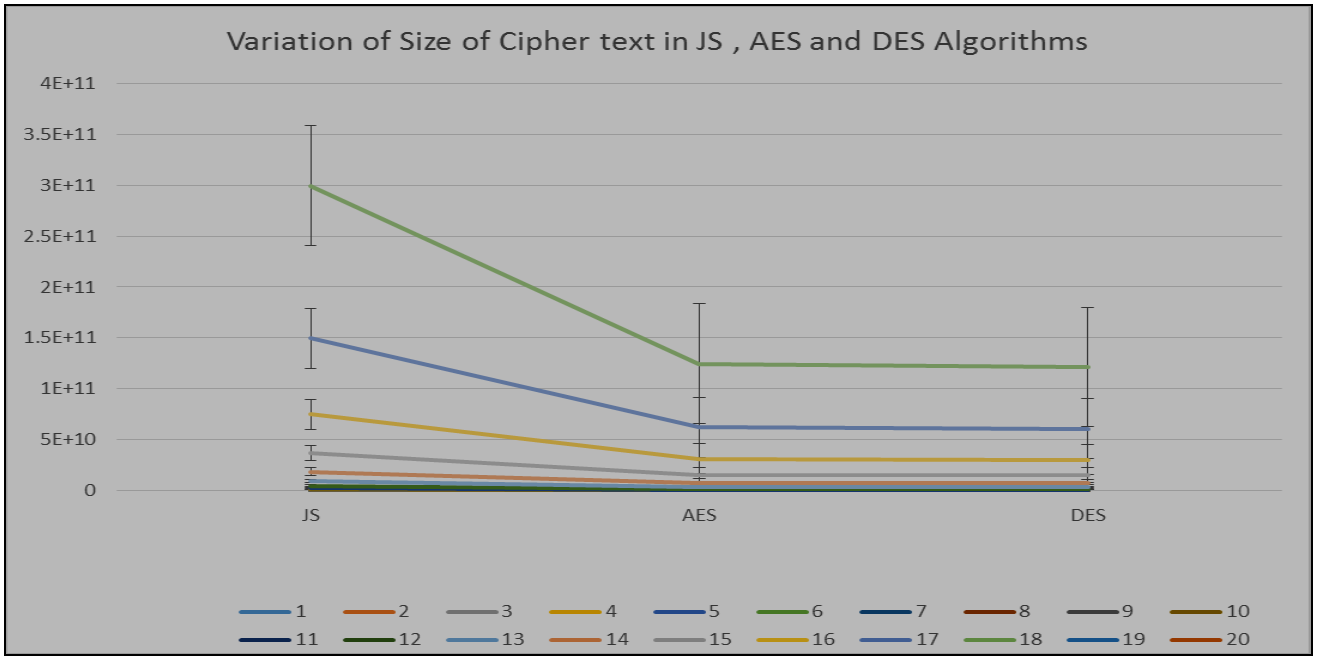

Fig. 4. Variation of size of cipher text in JS, AES and DES Algorithm

B. Inferences from the results:

Table 2 presents the statistical data for all three algorithms. From the statistical data, a bar graph is drawn. From the bar graph (figure 3) it is clearly seen that, the results of JS Algorithm are drastically high as compared to AES and DES algorithm.

Fig 4 shows the variations of texts in all three algorithms. The variation in case of JS Algorithm is continually increased whereas in case of AES and DES algorithm it is almost constant and uniform.

V. CONCLUSION

The research had drastically positive results in case of the cipher text sizes. The text data set was initially kept with 100 characters (consumes 200 bytes of plain text data), the increase in the plain text length was increased in quadratic manner. The total percentage of increase in cipher text with respect to current plain text was observed 106% (JS Algorithm), 16 %. (AES Algorithm), 13% (DES Algorithm). The statistics for all 30 text data set were almost same for AES and DES algorithm but it was continually varying for JS algorithm. The variation was observed due to the random nature of the algorithm. This concludes that JS algorithm is suitable for larger text files as well.

VI. FUTURE SCOPE

The future scope of this work is vast and feasible. The vital issues which cloud users and vendors generally face are data integrity, data confidentiality, and access control and data manipulation in the encrypted domain. Therefore, above algorithm can be turned into generic algorithm and can be used with cloud storage security. The problem statement of proposed research work is summarizing as:

- To develop unbreakable, robust and randomized encryption algorithm which is fruitful against any security attacks on cloud platforms.
- To develop generic encryption algorithm which can encrypt any type of multimedia file on cloud with efficient computational complexity.

REFERENCES

1. Applied Cryptography-Protocols, Algorithms and Source Code in C ,
2. Prathamesh Churi, Medha Kalelkar, and Bhavin Save, “JSH Algorithm: A Password Encryption Technique using Jumbling-Salting-Hashing”, International Journal of Computer Applications (0975-8887), Vol. 92-No.02, April 2014.
3. P. P. Churi, V. Ghate and K. Ghag, "Jumbling-Salting: An improvised approach for password encryption," 2015 International Conference on Science and Technology (TICST), Pathum Thani, 2015, pp. 236-242. doi: 10.1109/TICST.2015.7369364
4. M. Stamp. Information Security : Principles an practice, Wiley publications , ISBN-13 978-0-471-73848-0
5. "A Performance Comparison of Data Encryption Algorithms," IEEE Information and Communication Technologies, 2005. ICICT 2005. First International Conference, 2006-02-27, PP. 84- 89.
6. O. Goldreich, S. Goldwasser, and S. Micali, “How to Construct Random Functions”, J. ACM, Vol. 33, No. 4, 1986, pp 210-217.
7. G. S. Tamizharasi, B. Balamurugan and S. L. Aarthy, "Scalable and efficient attribute based encryption scheme for point to multi-point communication in cloud computing," 2016 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, 2016, pp. 1-4.
8. Mahalle, Vishwanath S., and Aniket K. Shahade. "Enhancing the data security in Cloud by implementing hybrid (Rsa & Aes) encryption algorithm." Power, Automation and Communication (INPAC), 2014 International Conference on. IEEE, 2014.
9. Maitri, Punam V., and Aruna Verma. "Secure file storage in cloud computing using hybrid cryptography algorithm." Wireless Communications, Signal Processing and Networking (WiSPNET), International Conference on. IEEE, 2016.
10. He, Xiang, Yihong Long, and Liheng Zheng. "A Transparent File Encryption Scheme Based on FUSE." Computational Intelligence and Security (CIS), 2016 12th International Conference on. IEEE, 2016.
11. Shahzad, Farrukh. "Safe haven in the cloud: Secure access controlled file encryption (SAFE) system." Science and Information Conference (SAI), 2015. IEEE, 2015.
12. Zhu, Shuaishuai, Xiaoyuan Yang, and Xuguang Wu. "Secure cloud file system with attribute based encryption." Intelligent Networking and Collaborative Systems (INCoS), 2013 5th International Conference on. IEEE, 2013.

13. Wang, Shulan, et al. "An efficient file hierarchy attribute-based encryption scheme in cloud computing." IEEE Transactions on Information Forensics and Security 11.6 (2016): 1265-1277.
14. El Houti, Youssef, and Andrea Miele. "Efficient update of encrypted files for cloud storage." Utility and Cloud Computing (UCC), 2015 IEEE/ACM 8th International Conference on. IEEE, 2015.
15. Mishra, Bharati, and Debasish Jena. "Securing Files in the Cloud." Cloud Computing in Emerging Markets (CCEM), 2016 IEEE International Conference on. IEEE, 2016.
16. Li, Cuixia, et al. "A video selective encryption strategy based on spark." Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2016 IEEE. IEEE, 2016.
17. Cui, Helei, Xingliang Yuan, and Cong Wang. "Harnessing encrypted data in cloud for secure and efficient image sharing from mobile devices." Computer Communications (INFOCOM), 2015 IEEE Conference on. IEEE, 2015.
18. Rad, Paul, et al. "A novel image encryption method to reduce decryption execution time in cloud." Systems Conference (SysCon), 2015 9th Annual IEEE International. IEEE, 2015

AUTHOR PROFILE

Mr. Prathamesh Churi is PhD research Scholar in Computer Science and Information Technology Symbiosis International (Deemed University), Pune, India. He is also Assistant Professor of Computer Engineering Department from SVKM's NMIMS Mukesh Patel School of Technology Management and Engineering, Mumbai, India. He is Associate Editor of International Journal of Advances in Intelligent Informatics (Indexed by Scopus), Indonesia. He has published more than 30 research papers in various international journals and conferences. He has been Co-Convener, Keynote Speaker, Session Chair and TPC Member of many reputed conferences at international level. His area of expertise includes Security and Privacy, Education Technology, Internet of Things. His email id is : Prathamesh.churi@gmail.com