International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-11, September 2020

A Dynamic Scalable Security Model for Data at Rest in fog Computing Environment

Tarek Husein, Aqeel Khalique, M. Afshar Alam, Bhavya Alankar

Abstract: ***Fog computing brings cloud services closer to the network's Edge. Despite various applications in today's world, these applications lack in data security aspects. Developers have few solutions that need to be tested thoroughly. Data encipherment is one of the most popular mechanisms to protect data confidentiality, data integrity, etc. We propose two steps flexible dynamic scalable model in which the system will dynamically choose an encryption mechanism depending on the access frequency of data being encrypted. If data is frequently accessed, then the model will choose the algorithm with minimum computational complexity. In next step, model will use a scalable approach to decide the security strength needed by determining size of encryption key. A longer key will be used to encrypt more sensitive and secretive data automatically by security model and a smaller key will be used to encrypt public data or less sensitive, saving the fog node from computation overload. Our model is more secure and dynamic in nature with scalable security strength.***

Keywords: Data Encryption, Scalable Security, Information Security, Fog Security.

I. INTRODUCTION

In the last few years, IoT devices are the new trend, it is estimated that in 2020, over more than 20 billion IoT devices will be connected to the network [1], this ever-expanding number of devices will generate an perpetually increasing size of data. IoT devices offer a wide range of services, which are in a way or another dependent or based on and derived from data. IoT devices require computational power to provide the functionality to all these services. At the same time, the functionality of IoT services depends on the speed of the data processing and decision making, whereas scalability and reliability might be an issue.

For Example, in a traditional Server-Client architecture, the data are processed in the server and consumed by the client, the overload of the server might affect the usability of many devices, making them useless. The future of IoT devices is forthcoming [2], smart systems using IoTs will be part of every aspect of our lives, smart transportation, smart health care systems, smart cities, smart houses, smart energy management, etc. examples of IoT applications include management, that enable a smart city. IoT networks give users and applications the ability to communicate through a heterogeneous distributed network at any time making countless services available. Cloud computing is a cost-effective solution to provide data processing, analyzing, and storing services. However, the centralized architecture of the Cloud network makes it difficult to respond effectively to some of the needs of IoT systems like location awareness, mobility, and low latency [3]. Cisco introduced the concept of Fog Computing as a solution to overcome the problems faced in the cloud environment by reducing the workload of cloud data centers and bringing cloud-like services closer to the edge of the network [4]. Fog computing uses a heterogeneous set of devices such as laptops, routers, smart phones, etc. distributed in different geographical locations as Fog nodes to provide low latency, location-aware wireless communication. Fog Computing is an architecture of decentralized computing for data storage and processing taking place in close proximity to the end-user and resulting in a reduction of data transmission to the cloud.

A. Fog Computing VS Cloud Computing

Cloud Computing can be described as a network of devices, servers, and computers, connected via the internet, used to store and process data, rather than using local computers or servers. Cloud Computing can be seen as a two-layer architecture as shown in Figure 1.

- The frontend or the end-user devices (computers, laptops, mobiles, tablets, ...etc.).
- The backend is located at a remote place from the end-user device and consists of data centers and processing systems.
- Storage and process capabilities: provide virtually unlimited storage and processing resources which can be integrated easily to store and process huge amounts of data.
- Low financial cost: compared to all the services it provides, surprisingly Cloud solutions are effectively less costly.

It also comes with some limitations, for example:

- Latency: due to long-distance communication between the end-user devices and the cloud, high latency might be an issue in the cloud environment.

Revised Manuscript Received on September 30, 2020.
***** Correspondence Author
Tarek Husein*, Department of Computer Science& Engineering, Jamia Hamdard, Delhi, India. Email: tarek.husein88@gmail.com
Aqeel Khalique*, Department of Computer Science& Engineering, Jamia Hamdard, Delhi, India. Email: aqeelkhalique@gmail.com
M. Afshar Alam, Department of Computer Science& Engineering, Jamia Hamdard, Delhi, India. Email: aalam@jamiahamdard.ac.in
Bhavya Alankar, Department of Computer Science& Engineering, Jamia Hamdard, Delhi, India. Email: balankar@jamiahamdard.ac.in

Data security and privacy: since the cloud network is connected via the internet, user private data will travel alongside millions of other user's data, which will make it a target for cyberattacks and data loss.

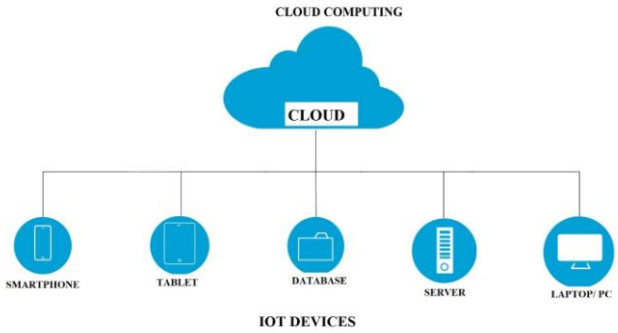

Fig. 1.Cloud Architecture

Fog Computing is an architecture of decentralized computing, where data storage and processing take place in close proximity to the end-user in Fog nodes as shown in Figure 2. The connection between the end-user devices and the fog nodes are almost instantly. Fog can be described as a middle layer between the cloud and the end-user, it controls which data should be sent to the cloud, and which data can be stored and processed in the fog nodes. It simply brings cloud-like capabilities to the edge of the network, and closer to the user.

Fog computing has many advantages, like:

- Latency: lower latency in communication, since Fog nodes are geographically closer to the end-user.
- Scalability: Scalability comes as an issue when handling a huge number of IoT devices in the cloud environment, although resources are virtually unlimited and effortless to add, it is difficult to obtain high network bandwidth. Contrarily, in Fog Computing Environment, multiple IoT devices are served by a single fog node, the scalability in network bandwidth can easily be satisfied by adding more nodes.
- Better user experience and Power-efficiency: downtime almost does not exist in fog, which provides better response and makes apps function faster, also fog computing adopts power-efficient protocols such as Bluetooth and Zigbee.

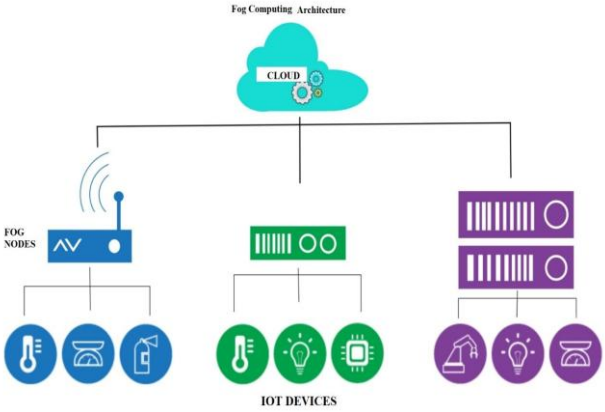

Fig. 2.Fog Architecture

Some of the limitations of Fog Computing are:

- More Expensive: companies adopting Fog in their business need to buy fog node devices (routers, gateways etc.)
- More complicated: Fog is an extra layer in the architecture if the network.

Table 1 shows a comparison between Fog and cloud computing:

Table- I: Fog and Cloud Computing Comparison

	Fog	***Cloud***
Hardware	Heterogeneous	General servers
Architecture	Distributed	Centralized
Mobility	Available	Not available
Latency	Lower	Higher
Scalability	High	Average
Computing capabilities	Lower	Higher
Number of nodes	Very large	Few

B. Scalable Security in Fog Computing Environment

For a better understanding of data security issues in Fog computing Environment, we can take a closer look at the security issues in Cloud Computing and other similar technologies. Since Fog Computing inherited many of the cloud characteristics, it also inherits many of its vulnerabilities.

Data encryption is one of the most popular techniques to protect data confidentiality.

Generally, data can be categorized into three different categories [5]:

- Data at rest: data stored in the storage unites
- Data in use: data currently being processed or consumed
- Data in transit: data transmitted through a communication network like the internet

The data confidentiality needs to be protected in a storage unit at the end-user device and the fog nodes (data at rest), also while communicating with other IoT devices, Fog nodes, and cloud (data in transit), and in the storage unit in the fog node (data at rest). In Fog Computing Environment, the available resource for computational operations are limited, and easily affected by the additional security measures [6]. Also, encrypted data requires more storage space and might affect the performance of the application on different devices and platforms. Scalable security can help us control the amount of resources allocated for securing the data depending on the nature of the data itself, in this approach the security strength will be determined automatically by the model depending on the nature of the data [5], different parameters can be taken in consideration, as File contents,

- Access Frequency,
- Time of Last access,
- Modification Frequency,
- Data Sensitivity

The rest of the structure of the paper is as follows; in section 2, we discuss some of the previous work in the field of data encryption in the fog environment.

In section 3, we present our problem statement. In section 4, we present a dynamic, flexible, and scalable model for securing data at rest in the Fog Computing Environment using three different cryptosystems.

In this model, the system has the flexibility to choose the encryption mechanism dynamically depending on the data access frequency without user intervention. In section 5, we present our implementation and compare the result of our experiment and in section 6, we conclude our work.

II. RELATED WORK

Fog Computing was first introduced by Cisco in 2014, it can be presented as a layer of Fog Nodes (router, switches, computers, mobile phone, smart edge device, car, temperature sensors, etc.) [7] that middles between the cloud and the end-users. Theses nodes are proximity closer to the end-user and perform some of the services and workload of the cloud, and as in cloud servers, they will also suffer from security problems, especially when the user stores important and secret data on the fog nodes [8]. Data Encryption can be a solution, but only taking into consideration the heterogeneity of Fog Computing Environment, and that usually Fog nodes have lower computational resources.

In research paper [9], the authors argue that AES is the most suitable Algorithm for data encryption in Fog storage nodes, for evaluating their work they used several parameters such as the CPU time of Encryption and decryption and the size of memory used. Although AES is adopted by the U.S. [10] government as a secure, efficient, and standard Encryption technique that use less hardware resource and requires smaller computations, which makes it suitable for data encrypting in Fog storage, this research should have compared AES against other encryption techniques, to give a clearer idea of its performance in Fog, they should have compared the performance over different key sizes (128, 192,256), also taking in consideration that Fog networks consist of heterogeneous devices, one algorithm might not be sufficient to cover all the cases. In research paper [11], the authors built a model for securing healthcare data in Fog Computing Environment, the model used pairing-based cryptography (PBC) for securing the data, model evaluation shows that the total cost of computation is high, leaving aside the cost of operations done by the server. In research paper [12], the authors argue that their approach of combining the fiestel cipher and the Substitution-Permutation (SP) networks is more effective in Fog Computing Environment, but one great weakness can be noticed in the mechanism of key generation, different key sizes are not supported, and the key is dependent on data itself. Another hybrid strategy was presented in research paper [13], this strategy combines steganography and cryptography, and is called emoticons. Emoticons proved to be simple and efficient in time cost, but it is not secure and can't stop usual attacks such as eavesdropping and IP spoofing. In research paper [14], the authors combined the advanced encryption standard (AES) with an improved implementation of RSA Algorithm, to propose a hybrid encryption algorithm. The algorithm was very efficient in terms of generating big primes. In research paper [15], the authors introduced a fine-grained scheme for data sharing in Fog Computing Environment, they explained that Attribute-Based Encryption (ABE), suffers from key-delegation, and they improved on it by using Ciphertext-Policy Attribute-Based Encryption CP-ABE to solve this problem, However, the biggest weakness of ABE, is that the computational time for the encryption and decryption is high, which makes it not suitable for Fog Computing Environment. In research paper [16], the authors also used encryption to secure Fog Computing Environment, by using Ciphertext-Policy Attribute-Based Encryption (CP-ABE) combined with signature algorithm, the required computation time in this strategy was high, making it unsuitable for the Fog Computing Environment. In the literature review for finding research gaps, we identified that none of the above-mentioned researches took into consideration the limited resources of Fog Computing Environment and the encryption/decryption process of data that will affect the performance of the fog devices.

III. PROBLEM STATEMENT

Security has always been one of the greatest challenges in almost all computing environments, especially when it comes to sensitive data. Security measures should be strong enough to ensure data confidentiality, however this comes with a cost a cost in resources and computation time. The need for data security is as important in Fog Computing Environment as in any other, but the limited resources in Fog Computing Environment stands against adding strong security measures. Scalable security offers a way of better usage of the limited resources in Fog Computing Environment, by simply stating that the strength of the encryption should be related to the nature of the data being encrypted, the more the data is sensitive and important the higher level of encryption strength should be used. In this paper, we propose a dynamic, flexible, and scalable model for securing data at rest in the Fog Computing Environment.

IV. DYNAMIC SCALABLE SECURITY MODEL

We have implemented 3 of the most popular cryptosystems namely RSA [17], GRSA [18], AES [19] with ECDH (Elliptic Curve Diffie–Hellman) [20] as a key exchange scheme. AES with ECDH has been introduced in the proposed dynamic scalable security model to provide additional security and flexibility to the model. Though RSA and GRSA have already been implemented, we have proposed a unique and hybrid combination of AES with ECDH to enhance the security strength of the proposed model. The difficulty in working with the Fog Computing Environment is that we need to always balance the usage of limited resources, and the computational cost of the encryption and decryption process. If fewer resources are allocated that would result in a higher computational time, which will make the main advantage of using Fog Computing Environment (low latency) irrelevant, also allocating more than needed resources by using higher security strength for not so important data may affect the performance of the network.

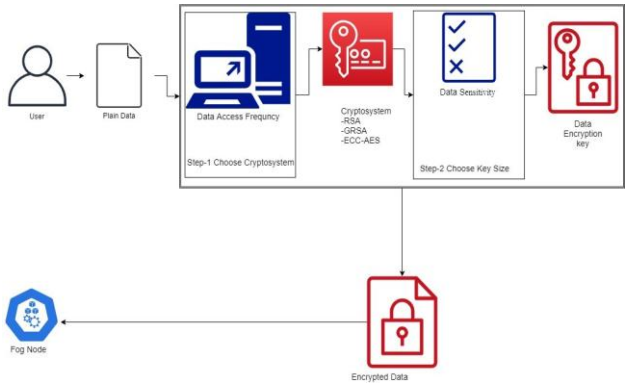

Fig. 3.A dynamic scalable security model in the Fog Computing Environment.

To overcome these problems, we propose a two steps dynamic security model, in which the system will dynamically assign a cryptosystem depending on the access frequency of the data being encrypted, if the data is frequently accessed the system will use the algorithm with minimum computational time. In the next step, the model will use a scalable approach to decide the security strength needed by setting the size of the encryption key. A longer key size will be used to encrypt more sensitive and secretive data automatically by the security model and a smaller key size will be used to encrypt public data or less sensitive, saving the fog node from computation overload. Figure 3 shows the dynamic scalable security model in Fog Computing Environment.

The encrypted data will be stored in the fog node. When a user request access to the encrypted data the fog node will decrypt it and send. To facilitate the decryption process, the model will add 1 Byte representing the encryption mechanism that was used in the encryption process o the end of the encrypted data. A better solution is to save this information in a safe storage space rather than within the data.

V. IPLEMENTATION AND RESULTS

For implementing the proposed model, we used java library functions of BigInteger class from the Java.Security and Java.Crypto libraries, in both RSA and GRSA [21], and the System.Security.Cryptography library [22] from "Microsoft .Net Framework" for implementing both AES with ECDH. All experiments were done on A laptop with intel core i5-8250U CPU @ 1.6 GHZ 6 MB cache and 16 GB of RAM.

A. RSA

RSA is an asymmetric cryptosystem. It was first introduced by Ron Rivest, Adi Shamir, and Leonard Adleman in the year 1977 [17]. In RSA algorithm, the security strength and the size of the encryption key are proportionally related, that means if we want to increase the security strength, we need to increase the size of the key.

The strength of the cryptosystem is determined by the time and resources needed to cryptanalysis it, the time need to acquire the original plain message or decrypt it. A comparison of the Security strength for different key sizes in RSA and ECDH cryptosystem is shown in Table 2.

Table- II: Fog and Cloud Computing Comparison

RSA Key Size (Bits)	*ECDH Key Size (Bits)*	*Time to break (MIPS)*
1024	160	10^ 12
2048	224	10^ 24
3072	256	10^ 28
7680	384	10^ 47

RSA is based on the integer factorization assumption, and it involves the following steps: key generation, encryption, and decryption [23].

A comparative table is shown in Table 3 to present time consumption for key generation, encryption and decryption using different key sizes.

Table- III: Computational Cost for different Key lengths in RSA

Key Size (bits)	*Key generation (MS)*	*Encryption (MS)*
512	34	100
1024	76	114
2048	286	240
4096	554	690

B. GRSA

In RSA, key generation is done using 2 prime numbers, in GRSA [24], the number of primes is determined by the user in the form of 2^k, which makes it more secure and way more difficult to cryptanalysis.

GRSA introduces Scalability to the Standard RSA algorithm, the user also can choose the size of the key that is going to be used (i.e. 128, 256, 512, 1024, 2048, ... bits). Tables 4, 5, 6 shows key generation times and encryption time for multiple key sizes using k = 3,4,5.

Table- IV: Computational time for different key sizes in GRSA using k = 3

Key Size (bits)	*Key generation (Milliseconds)*	*Encryption (Milliseconds)*
256	106	3
512	322	8
1024	987	23
2048	4877	73
4096	28248	270

Table- V: Computational time for different key sizes in GRSA using k = 4

Key Size (bits)	*Key generation (Milliseconds)*	*Encryption (Milliseconds)*
256	322	5
512	1205	12
1024	6254	36
2048	35475	102
4096	322480	385

Table- VI: Computational time for different key sizes in GRSA using k = 5

Key Size (bits)	*Key generation (Milliseconds)*	*Encryption (Milliseconds)*
256	1575	7
512	7652	18
1024	42854	60
2048	314854	192
4096	512560	784

C. AES with ECDH

We used Elliptic Curve Diffie–Hellman [20] as a key exchange protocol to generate and exchange a shared secret, this shared secret is then used then as an encryption key in AES algorithm.

In Diffie-Hellman protocol, no previous contact between the sender and the receiver is needed.

Every party will generate its pair of private and public keys.

Then public keys will be exchanged or broadcasted through the communication network.

To get the shared secret each party (the sender and the receiver) will use its own private key combining it with the other party public key.

The steps can be seen in the following algorithm:

1. Sender and Receiver choose their private keys K_s, K_R randomly.
2. Sender and Receiver calculate their public key S=K_sP, R =K_RP, and then exchange the keys among each other.
3. Sender and receiver now calculate the shared secret $Q = K_s(K_R\ P) = K_R(K_sP) = KP$, the shared key is K_SK_R.

In the next step, we will use the shared key, as an encryption key in Advanced Encryption Standard (AES) [19], AES uses keys of 128, 192, 256 bits of length to encrypt and decrypt 128 bits of data.

Table 7 shows the computational cost for key generation in ECDH, and encryption time of AES using 128, 192, 256 keys sizes.

Table- VII: Computational cost of key generating and data encryption time of AES -ECDH using different keys sizes

Key Size (bits)	*Key generation (Milliseconds)*	*Encryption (Milliseconds)*
128	1	5
192	2	7
256	2	8

VI. RESULTS AND ANALYSIS

In this section, we test our model through different scenarios.

In Figure 4, we show a comparison of the time cost for key generation for different key lengths. AES-ECDH proves to be the best but with a slight difference.

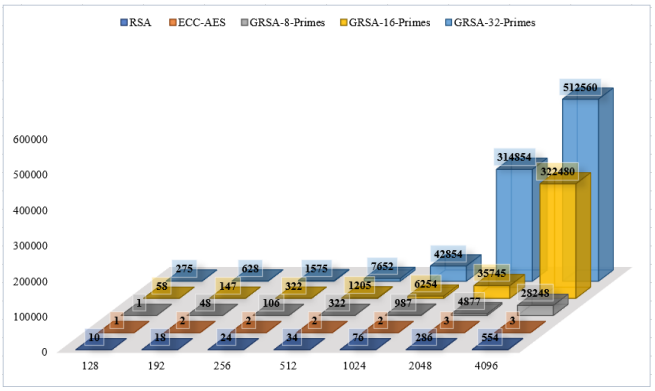

Fig. 4.Key Generation Time Comparison

Figure 5 shows the time need for encryption of text file of size 16 KB using different key sizes.

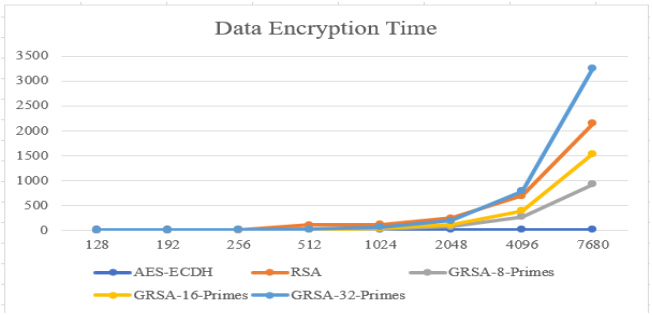

Fig. 5.Data Encryption Time on 16 KB file

In Figure 6 we used three files of 1MB, 2 MB, 3MB using a key size of 2048 in RSA and GRSA, and a key size of 256 in AES-ECDH.

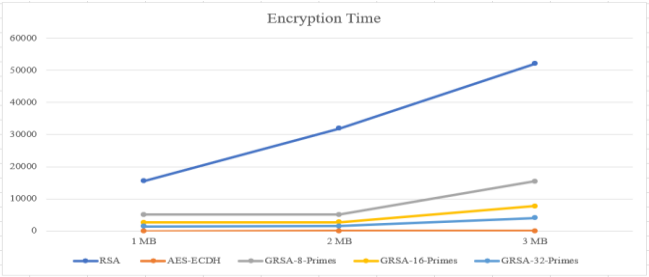

Fig. 6.Encryption Time Comparison of different file sizes

In Figure 7, we compare the memory size used in Encryption using all three algorithms on 1 MB file, using a key size of 2048 in RSA and GRSA, and a key of 256 in AES-ECDH.

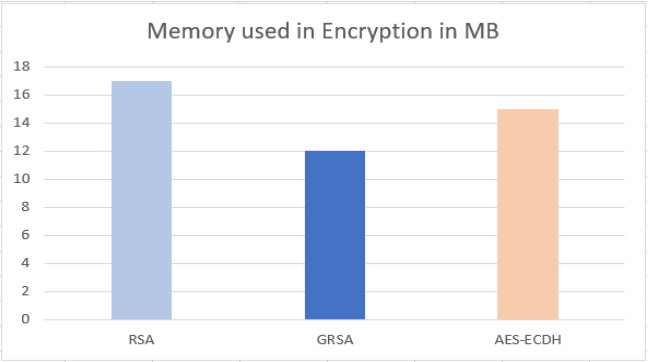

Fig. 7.Memory usage for encrypting file of size 1 MB

VII. CONCLUSION

In this paper, we have presented a dynamic scalable model for securing data at rest in the Fog Computing Environment. We have implemented three data cryptosystems (RSA, GRSA, AES-ECDH) making the model hybrid in nature. The model will dynamically choose one of the three cryptosystems depending on the data access frequency, and then security strength represented in key size will be determined by the sensitivity of the data. The scalability in this approach will help us avoid the unnecessary usage of limited resources in the Fog node and minimize the computational time by using the suitable cryptosystem based on the data access frequency. The model was implemented using Java.crypto, Java.Security and Security.Cryptography from .Net Framework and results have been presented. The analysis of the model shows the time and memory consumption using different key sizes and different scenarios. Also, the security strength of the proposed model will enhance due to the dynamic selection of cryptosystem, scalable key size determined by the data sensitivity and its hybrid nature. It will be computationally infeasible for any intruder to determine at least one or more parameters without any knowledge of the cryptosystem being used and the key size.

REFERENCES

1. (2016) Up 31 Percent From. Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017. Retrieved from https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016.
2. Yan, Z., Niemi, V., and Yang, L.T. (2016) Key technologies for 5G, the next generation of mobile networks and services. International Journal of Communication Systems, 29 (16), 2328–2329, doi:10.1002/dac.3226.
3. Rahman, G. and Wen, C.C. (2018) Fog Computing, Applications, Security and Challenges, Review. International Journal of Engineering & Technology, 7 (3), 1615–1615, doi:10.14419/ijet.v7i3.12612.
4. Stojmenovic, I., Wen, S., Huang, X., and Luan, H. (2016) An overview of Fog computing and its security issues. Concurrency and Computation: Practice and Experience, 28 (10), 2991–3005, doi:10.1002/cpe.3485.
5. Khalique, A., Hussain, I., Alam, M.A., and Khan, T.A. (2020) Scalable Security Based on Data Classification Using Generalized RSA in Cloud Storage, in Proceedings of ICETIT 2019, Springer, pp. 973–983
6. Shmueli, E., Vaisenberg, R., Elovici, Y., and Glezer, C. (2010) Database encryption: an overview of contemporary challenges and design considerations. ACM SIGMOD Record, 38 (3), 29–34C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.
7. Hoque, S., Brito, M.S.D., Willner, A., Keil, O., and Magedanz, T. (2017) Towards container orchestration in fog computing infrastructures. 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), 2, 294–299.
8. Yi, S., Li, C., and Li, Q. (2015) A survey of fog computing: concepts, applications and issues. Proceedings of the 2015 workshop on mobile big data, pp. 37–42.
9. Vishwanath, A., Peruri, R., and He, J.S. (2016) Security in Fog Computing through Encryption. International Journal of Information Technology and Computer Science, 8 (5), 28–36, doi:10.5815/ijitcs.2016.05.03.
10. Miller, F.P., Vandome, A.F., and Mcbrewster, J. (2009) Advanced Encryption Standard. Alpha Press.
11. Hamid, H.A.A., Rahman, S.M.M., Hossain, M.S., Almogren, A., and Alamri,A. (2017) A Security Model for Preserving the Privacy of Medical Big Data in a Healthcare Cloud Using a Fog Computing Facility With Pairing-Based Cryptography. IEEE Access, 5, 22 313–22 328, doi:10.1109/access.2017.2757844.
12. Usman, M., Ahmed, I., Aslam, M. I., Khan, S., & Shah, U. A. (2017). SIT: a lightweight encryption algorithm for secure internet of things. arXiv preprint arXiv:1704.08688.
13. Kumar, H., Shinde, S., and Talele, P. (2017) Secure fog computing system using emoticon technique. International Journal on Recent and Innovation Trends in Computing and Communication, 5 (7), 801–808.
14. Liang, C., Ye, N., Malekian, R., and Wang, R. (2016) The hybrid encryption algorithm of lightweight data in cloud storage. 2016 2nd International Symposium on Agent, Multi-Agent Systems and Robotics (ISAMSR), pp. 160–166.
15. Jiang, Y., Susilo, W., Mu, Y., and Guo, F. (2018) Ciphertext-policy attribute-based encryption against key-delegation abuse in fog computing. Future Generation Computer Systems, 78, 720–729, doi: 10.1016/j.future.2017.01.026
16. Alrawais, A., Alhothaily, A., Hu, C., Xing, X., and Cheng, X. (2017) An Attribute-Based Encryption Scheme to Secure Fog Communications. IEEE Access, 5, 9131–9138, doi:10.1109/access.2017.2705076
17. Rivest, R.L., Shamir, A., and Adleman (1978) Method for obtaining digital signatures and public key cryptosystems. Commun. ACM, 21 (2), 120–126.
18. Lone, A.H. and Khalique, A. (2016), Generalized RSA using 2kprime numbers with secure key generation, doi:10.1002/sec.1619
19. Standard, A.E. (2001) Federal information processing standard (fips) publication.197. National Bureau of Standards, Washington, DC.
20. Bernstein, D.J. (2006) Curve25519: new Diffie-Hellman speed records, in International Workshop on Public Key Cryptography, Springer, pp. 207–228.
21. (2018).URL https://docs.oracle.com/javase/7/docs/api/java/math/class-use/ BigInteger.html#java.security.interfaces.
22. Dotnet-Bot.URLhttps://docs.microsoft.com/en us/dotnet/api/system.security. cryptography.
23. Zhang, X. (2003) Theory and techniques of digital signature, Mechanic Industry Press, Beijing.
24. Lone, A.H. and Khalique, A. (2016), Generalized RSA using 2kprime numbers with secure key generation, doi:10.1002/sec.1619.

AUTHORS PROFILE

Tarek Husein currently pursuing M.Tech in Computer Science Engineering from Jamia Hamdard, New Delhi.

Aqeel Khalique completed his M.Tech. in CSE from IIT Roorkee. Aqeel has several publications in domain such as Information Security, Pervasive Computing, IoT, Scalable Security, Sustainable Computing etc. Aqeel is currently working as Assistant Professor in Dept. of CSE, Jamia Hamdard, New Delhi. Aqeel is in Editorial Board of International Journal of End-User Computing and Development (IJEUCD). Aqeel is also member of many research bodies such as ISTA, ACM etc.

Prof. M. Afshar Alam is Head and Dean, Department of Computer Science and Engineering, School of Engineering and Technology, Jamia Hamdard, New Delhi. He received the Ph.D. degree in Computer Science from Jamia Millia Islamia. His research areas include Software Re-engineering, Data Mining, Bio-Informatics, Fuzzy Databases, and Sustainable Development. In his 25 years of experience in teaching and research, he has authored 9 books and has more than 150 research papers in reputed journals to his credit. He is also member of many government bodies including UGC, AICTE and many more.

Bhavya Alankar is working currently as Assistant professor in Department of Computer Science and Engineering, School of Engineering and Technology, Jamia Hamdard of at Jamia Hamdard. He has more than 14 years of experience in teaching and research. His research areas include VLSI Design, Software engineering, Machine learning.