IJCSE International Journal of Computer Sciences and Engineering Open Access
Survey Paper Vol.-7, Issue-6, June 2019 E-ISSN: 2347-2693

A Survey on Assured Data Deletion in Cloud Storage

Seema B. Joshi[1*], Shaileshkumar D. Panchal[2]

[1]Department of Cyber Security, Gujarat Technological University, Ahmedabad, Gujarat, India
[2]Information Technology Department, Vishwakarma Government Engineering College, Ahmedabad, Gujarat, India

[*]*Corresponding Author: ap_seema@gtu.edu.in, Tel.: +09898180113*

DOI: https://doi.org/10.26438/ijcse/v7i6.548553 | Available online at: www.ijcseonline.org

Accepted: 10/Jun/2019, Published: 30/Jun/2019

***Abstract*—** With the rapid growth of cloud computing technology, more and more users store and share their data through cloud storage. Major concern is the inadvertent exposure of sensitive data of potential cloud users. Sometimes the cloud server may not delete the data honestly for financial intensives so that data deletion becomes a security challenge. Sometimes unintended disclosure leads to heavy financial penalties and reputational damage. The traditional approach to this problem is encryption of the data before outsourcing and destruction of the encryption key when detecting. Moreover, most of the existing methods can be summarized with the one-bit-return protocol. In which, cloud storage server deletes the data and returns one-bit as a result either 0 or 1 means failure/success. Sometimes this result misguides the user, but user has to believe the returned result because user can not verify it. As userslose their direct control over their data in cloud storage. Hence, assured data deletion is highly required in cloudstorage. In this paper, we aim to analyze assured deletion methods for the cloud, identifying the cloud features that pose a threat to assured deletion and described various assured deletion challenges.

***Keywords*—**Assured data deletion, User assurance, Cloud storage, Cloud security.

I.INTRODUCTION

Cloud computing is the mixture of the parallel computing, distributed computing and grid computing. It connects configurable devices, large scale storage and computing resources together through the Internet. With a ubiquitous and convenient computing environment, cloud users can use various data services such as cloud storage service, outsourcing computing service, on-demand self-service, etc. [1]. In the cloud storage, the resource-constraint users can outsource the expensive storage into the remote cloud and experience plentiful storage services.

For cloud computing users, a major concern is disclosure of sensitive data. The security issues arise from insecure to incomplete data deletion and exploitation of virtualization due to side channel attacks and various other types of attacks is investigated through research. The unintentional disclosures of tenant's sensitive information costs are high and include monetary losses for customers and providers and loss of reputation. In 2019, as per the Thales data threat report, it is identified that 97% of respondents use sensitive data on digitally transformative technologies. 86% of respondents said they are vulnerable to data security threats and predicts that this share will rise; data leakage has become one of the important issues that affect the development and application of cloud computing, where insecurity deletion of data is a major cause of data leakage [2].

However, the assured deletion guarantees in the cloud are important for tenants' perspective as well as cloud service provider perspective. It is essential to receive assurances for data deletion as agreed for the tenant. Such guarantees are needed to comply with data regulations of various countries and regions and also address tenants' requirements and expectations which are important for cloud service providers. Furthermore, data deletion assurances can also become a valuable in the market for a cloud provider.

The primitive of secure data deletion has been widely studied in the past decade [3, 4, 5, 12]. It is important to note that in most of the existing data deletion methods can be summarized same protocol i.e. "one-bit-return" protocol. In this protocol, the data owner sends a command to delete data from physical storage medium and then receives a one-bit reply (i.e. Success/Failure) which indicates the result of the deletion operation. For example, operating system gets deletion result of one bit by removing the link. The deletion results can be misleading the data owner because the content of the file still remains on the disk, attackers can recover the file by scanning the disk.

The cloud user has no access to the infrastructure; henceforth it is difficult for them to verify deletion of data from the cloud. For the cloud service providers, there are number of significant features such as multi-tenancy, virtualization, service delivery models, scalability, high availability and data backup, etc. all of which pose various challenges with regards to provide data deletion assurances.

Here, we have identified three adversarial models to context the challenges of assured deletion in the cloud. One involves a distrusted cloud server, second is related with the semi trusted cloud server and the third is trusted cloud server. We present requirements for assured deletion and public verification for a tenant for the distrusted cloud server based adversarial model and then analyse existing methods for assuring deletion in such scenarios. We identified the limitations of such solutions. Afterwards, using the semi trusted cloud server model we draw requirements for assured deletion and public verification in such a context.
In summary, our main contributions are as follows

- We discuss the issue of assured deletion in the cloud from three perspectives, the distrusted; semi trusted and trusted cloud server scenarios, giving a distinctive mapping between requirements and challenges.
- We identify essential cloud features which present challenges to assured deletion and public verification, and offer a systematic analysis and discussion of these challenges for assured deletion for both cloud tenants and providers.
- We discuss a comprehensive study of the existing solutions and identifying limitations and challenges in the area.

The rest of this paper is organized as follows. Section II introduces assured deletion and presents the three adversarial models discussed in this paper. Section III discusses the requirements of assured deletion and analyse the existing methods. Section IV presents the limitations of the existing methods and Section V is about the conclusion of paper.

II. ASSURED DATA DELETION

Data deletion is one of the important aspects of managing and security sensitive data in the context of data security. Partial or incomplete data deletion may lead to inadvertent exposure of users' sensitive data. Assured data deletion is the key element in assuring confidentiality. It is achieved when deleted data is permanently inaccessible to anyone and assurance is given to the tenants that data is securely deleted with proof of verification [7].

A. Adversarial Models

In the following subsections, we describe three adversarial models. In the first model, we consider a scenario where a tenant uses the services of a distrusted cloud server while the second model considers a scenario where the cloud server is semi trusted and in the third model trusted cloud server is considered. In all three scenarios, we assume that cloud tenants desire to have their data assuredly deleted from the cloud infrastructure. The first and second model is considered and acts as an adversary while the third cloud server is considered as trusted cloud server so it is not an adversary and it is ready to provide an assured deletion as a service.

- **Distrusted cloud server based adversarial model**

It is considered a situation where a public cloud server is used by cloud tenant for storage. The cloud tenants outsource most of their data to the cloud but they are suspicious about the data disposal process of cloud storage. Here, it is assumed that the cloud tenant is aware about the risks of incomplete or impartial data deletion. The cloud tenant desires to ensure that even after deletion; data will remain safe without any extra cost. It is also assumed that the cloud server is curious about tenants' data and it has some other malicious tenants who are interested in other tenants' data. The malicious tenant may request more resources such as processing server during provisioning of services. The malicious tenant investigates the provided resource for sensitive data that may have been left behind by a previous tenant before writing any data to the availed resource.

- **Semi trusted cloud server based adversarial model**

In this scenario, we assume that the cloud server is a "semi trusted" server. That means the cloud server may not follow our presented scheme truthfully to delete the data but return an error deletion result to mislead the cloud tenant for financial incentives. Moreover, the tenant is assumed to be trusted and the communication channels are assumed to be secure. The tenant does not maintain any copy locally, so he/she would try to download the data while he/she need the data. Here, two types of attacks are considered. First, the semi trusted cloud server may delete some data randomly which are rarely accessed by the respective cloud tenants for economic interests. Second, the semi trusted cloud server may not delete the data as per the respective cloud tenant's request and returns an error result to mislead the him/her for benefits.

- **Trusted cloud server based adversarial model**

For our third model, we consider a scenario where a cloud server is trusted but prone to accidental data leaks due to incomplete deletion. We assume that the cloud server precisely manages the security mechanisms for tenant's data protection and does not have any intentions to leak tenant's data. Additionally, cloud service provider is interested to

provide confidentiality and comply with legal and standard regulations based on service level agreement. In spite of the trusted environment of cloud server, malicious tenants may arbitrarily probe their resources for partially deleted data. The malicious attackers may target decommissioned machines to steal data from the cloud server unless data is completely deleted from the cloud storage. In the next section, we review the requirements for assured deletion, existing approaches and limitations with respect to the above three scenarios.

III. REQUIREMENTS FOR ASSURED DATA DELETION IN CLOUD STORAGE

In the situation, when cloud tenant does not trust the cloud server following are the requirements to be considered for assured data deletion.

- **Fine-grained:**
  Cloud tenant's data deletion should be fine-grained in which only the target data is deleted while remaining data should be safe and accessible. Fine-grained deletion gives more user control to delete the data in the cloud storage and therefore, deletion operations cost is reduced.

- **Availability of services:**
  Assured deletion should be easy and it should not be affected tenant's daily work and productivity. The other service availability of cloud server should not be affected due to assured deletion operations.

- **Cloud Computation:**
  Cloud tenant should continue to work with data without any problems. Tenants should be able to complete necessary data operation related to computation such as searching and sorting.

- **Complete deletion:**
  Assured deletion should be performed to delete all copies of data associated with the deleted data including the metadata with assurance of complete deletion.

- **Timeliness:**
  Deletion should be completed punctually without overhead; deleted data should be inaccessible from the environment immediately after deletion is complete.

- **Error Handling:**
  Assured deletion should have error handling functionalities so that it can be completed without any error within time limit.

- **Acknowledgement of deletion:**
  Cloud tenant should be acknowledged for assured deletion operations after completion.

It is always challenging to assured deletion in the cloud storage for the cloud tenants because they have no control over cloud infrastructure and geolocation of cloud data storage. Cloud tenants have to believe that their data is to be deleted securely from the cloud server based on his/her request. In the next section, we discuss the existing approaches to assured deletion and their limitations.

IV. EXISTING DELETION METHODS AND LIMITATIONS

In this section, we present existing approaches to guarantee deletion in the cloud. With the discussion of existing approaches and its limitations, we aim to outline the requirements for assured data deletion in the cloud storage.

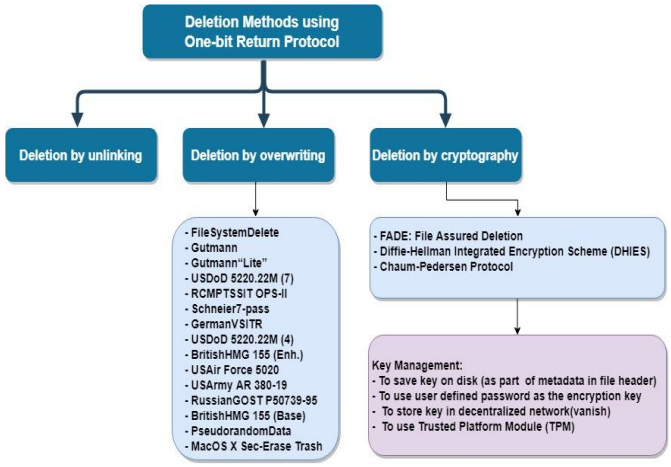

Figure 1. Data deletion methods using one-bit return protocol

- **Deletion by unlinking**

The extensive study of the secure data deletion primitives has been done in past decade [1, 7, 9]. Essentially, the "one-bit-return" protocol is summarized in most of the existing data deletion methods. That is, the user sends a command to delete data from physical storage medium, and then receives a one-bit reply of either success or failure that indicates the result of the deletion operation. For example, while user hitting the 'delete' button to delete a file, the operating system removes the link of the file from the underlying file system and returns one bit to the user: 'Success'. The return of the 'Success' bit can be misleading.

- **Limitation:**
  The system only deletes the link of the file; however, the content of the file remains on the disk. Attacker can recover the file by scanning the disk [9]. So, with this limitation deletion by unlinking is not adequate solution in real time applications.

- **Deletion by overwriting**
  Many protocols have been proposed to delete the file contents. Overwriting technology is applied to design secure data deletion schemes by many researchers. Overwriting techniques are used to delete the content of the file by overwriting with random data [3, 4, 8, 13, 14]. In general, the secure data deletion problem has been hypothetically solved by overwriting the storage medium.

  - **Limitation:**
    Most of the overwriting method cannot support verification. However, some scheme are supported verification, but with the help of trusted third party.

    The other inherent limitation is that the proposed protocols are much inefficient for practical applications.

- **Deletion by cryptography**

  The first cryptography-based solution for secure data deletion problem was presented by Boneh and Lipton in 1996 [7]. In this proposed solution, they encrypt all the data before saving it and then delete the plaintext. Later, the decryption key is deleted to make the ciphertext invalid.

  The large amount of data can be deleted using the cryptography-based solution. Particularly in distributed storage, all the duplicate copies of the data that are backed up in distributed locations can be deleted at once time.
  However, data owner also can not verify the result of the deletion operation in those methods. Data owner has to trust the returned result. Moreover, the ciphertext is stored in the physical medium which may create a data leakage threat. Therefore, it is necessary to find out publicly verifiable data deletion protocol.

In Perlman's concept [8], the data is encrypted before outsourcing then encryption key is deleted, so that the encrypted data is become unrecoverable after the deletion in the cloud storage.
Particularly, the data owner first encrypts the data file with a randomly generated data key and then the third-party key manager i.e. ephemerizer further encrypts the data key with a control key that is corresponding to the data file.

The control key is time-based and will be automatically destroyed when the predefined time for the data file expired or on deletion request from the owner. In Perlman's concept, the assured deletion problem is converted into key management problem.

- **Limitations:**

The ciphertext is still stored in the physical medium.

Data owner is dependent on third party key manager.

Key management problem in Perlman's concept of one-bit return protocol.

The public verifiability is not available for the result of the deletion operation.

- **Key Management Methods and its limitations:**

The key management becomes essential when cryptography is used for data deletion problem. There are several approaches proposed to manage cryptographic keys in the past literature.

- **Key Management Method 1:**

The first method is to just save the key on the disk, along with the encrypted data. Typically, it is stored as part of the meta data in the file header [7, 8, 12, 13, 14, 15].

- **Limitation:**

In this key management approach, deleting the data involves overwriting the disk location where the key is stored. The ciphertext becomes useless immediately, once the key is erased.

This method quickly erases the data by just removing the small block of data (AES-128 bit key) needs to be overwritten, but it may not give assurance about data deletion.

In the contrast, once the key is restored, the deleted data can be fully recovered by attacker. This method may degrade the security.

- **Key Management Method 2:**

The second method for key management is to use a user-defined password as the encryption key. The key is derived on the fly in RAM upon the user's entry of the password so it is never saved on the disk.

- **Limitation:**

The passwords are naturally bounded by low entropy (typically 2030 bits). Hence, the brute-force attack is possible

to identify cryptographic keys which are derived from passwords.

- **Key Management Method3:**

The third method is to store the key in a decentralized network. The method name is called Vanish, which is proposed by Geambasu. A random key is generated to encrypt user's data locally and then the key using Shamir's secret sharing scheme is to be distributed to peer-to-peer, distributed hash tables (DHTs). This method is called vanish in which the shares of the key naturally disappear, due to the fact that the DHT is constantly changing.

o **Limitation:**

The sybil attacks is performed to recover the stored key before it vanish [9]. The original Vanish scheme cannot guarantee to secure deletion of the key.

- **Key Management Method 4:**

The fourth method is to store the key in a tamper resistant hardware module (e.g., TPM) and define the Application Programming Interface (API) to manage the stored keys.

o **Limitation:**

This design follows the one-bit return protocol, user has to trust upon the correct implementation of the software inside the trusted platform module.

However, Perlman's concept-based schemes have three serious problems:

- Firstly, heavy computation in encryption of data before outsourcing in the user side.
- Secondly, the encrypted data remains in the cloud server after deletion operations. It may be leaked user's sensitive data or possibility of brute force attacks using powerful machines like quantum computer.
- Third problem is, the cloud computing performance on the outsourced data is difficult because of encryption. However, the main goal of cloud computing is computation of outsourced data.

Another cryptography-based approach is FADE to assure deletion [13]. It supports policy-based assured deletion. When associated file's access policies are revoked, assured deletion is achieved. It allows tenants to revoke policies of the target file or data that needs to be deleted, so that fine-grained deletion is achieved in this approach.

o **Limitation:**

The FADE approach is not considered for multiple files existence, which is the normally required for cloud setup.

The FadeVersion is the extension of FADE approach, in which multiple files are allowed [7].

o **Limitation:**

The third party is supposed to manage the increased number of encryption keys. So, data owner has to trust upon third party.

Another approach for verifiable data deletion scheme was proposed in 2010, which called "Proof of Erasability" (PoE) [16]. Besides, to delete data from the embedded devices, Perito and Tsudik proposed a similar scheme, which called "Proofs of Secure Erasure" (PoSE-s). These two schemes follow same pattern for overwriting mechanism and as a deletion proof.

In 2016, a Trusted Platform Module (TPM) based publicly verifiable data deletion scheme is proposed [9]. They combine Chaum-Pedersen Zero Knowledge Proof with Diffie-Hellman encryption protocol to realize data confidentiality and data provable deletion.

Recently, the Blockchain-based publicly verifiable data deletion scheme is proposed to reach public verification without any trusted third party in 2018.

In the provable data transfer scheme, they delete the transferred data by revoking the decryption key, and verify the integrity of the transferred data on the new cloud.

Another provable data transfer protocol, which can enable the data owner to migrate the outsourced data between different cloud servers, and verify the data integrity on the new cloud [26]. Finally, the original cloud server deletes the transferred data and returns a deletion proof.

The scheme for secure outsourced data transfer anddeletion with public verification is proposed in 2018. The author has introduced homomorphic encryption and homomorphic authenticator to realize verifiable deletion and proof data possession. The blockchain based public verification of data deletion scheme is proposed in 2018 [27].In this method based on trust-but-verify principle data owner can verify the result of data deletion using blockchain technology.

V. CONCLUSION

In this paper, we have stated the importance of assured data deletion in the cloud storage. We have surveyed the existing solutions of data deletion against requirements and outlined the limitations in cases of different adversarial modelsof cloud server.Assured data deletion is a significant obstacle for adopting public clouds services. The conventional assumptions about data deletion are either trust or distrust. The data owner has to believe the returned result because he/she cannot verify it. It is important to allow the data owners to control and verify how their data is handled. Moreover, it is reviewed that many researchers have

proposed new solutions for assured data deletion in the cloud storage, that is significant for public cloud tenants and it provides an essential path to a wider community of researchers to extend this work to providedata confidentiality in cloud storage.

REFERENCES

[1] P. Mell and T. Grance, *"The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology"*, NIST Special Publication, Vol. 145, pp. 7, 2011.

[2] G. Edition, *"The Changing Face of Data Security 2019 Thales Data Threat Report"*, 2019.

[3] P. Gutmann, *"Secure Deletion of Data from Magnetic and Solid-State Memory"*, 6th USENIX Security Symposium Proceedings, pp. 22-25 July, 1996, San Jose, California, 1996.

[4] Mather, Tim and Kumaraswamy, Subra and Latif, Shahed, *"Cloud security and privacy: an enterprise perspective on risks and compliance"*, O'Reilly Media, Inc., Sebastopol, CA, 2009.

[5] Garfinkel, S.L., Shelat, A., *"Remembrance of data passed: A study of disk sanitization practices"*, IEEE Security and Privacy, Vol. 1(1), pp. 17-27, 2003.

[6] T. Waizenegger, F. Wagner, *"SDOS: Using Trusted Platform Modules for Secure Cryptographic Deletion in the Swift Object Store"*, Proc. 20th International Conference on Extending Database Technology (EDBT), ISBN 978-3-89318-073-8, pp. 550–553, 2017.

[7] K. M. Ramokapane and J. M. Such, *"Assured Deletion in the Cloud: Requirements, Challenges and Future Directions"*,Proceedings of the 2016 ACM on Cloud Computing Security Workshop, pp. 97-108, NY, USA, 2016.

[8] Y. Luo and D. Wang, *"Enabling Assured Deletion in the Cloud Storage by Overwriting"*, Proceedings of the 4th ACM International Workshop on Security in Cloud Computing,pp. 17–23, NY, USA, 2016.

[9] F. Hao, D. Clarke, and A. F. Zorzo, *"Deleting Secret Data with Public Verifiability,"*IEEE Transactions on Dependable and Secure Computing, vol. 13, Issue. 6, pp. 617-628, 2015.

[10] S. Renuga, S. S. K. Jagetheeshwari, "*Efficient Privacy-Preserving DataSanitization over Cloud Using Optimal GSA Algorithm"*,The Computer Journal, Vol. 61, Issue 10, pp. 1577–1588, 2018.

[11] B. Chen, S. Jia, L. Xia, and P. Liu, *"Sanitizing Data is Not Enough! Towards Sanitizing Structural Artifacts in Flash Media"*,Proceedings of the 32nd Annual Conference on Computer Security Applications, ISBN: 978-1-4503-4771-6,pp. 496-507, California, USA, 2016.

[12] C. Y. B, J. Wang, X. Tao, and X. Chen, *"Publicly Verifiable Data Transfer and Deletion Scheme for Cloud Storage"*, Information and Communications Security, Springer International Publishing, pp 445-458, Vol. 11149, 2018.

[13] C. Cachin, *"Policy-based Secure Deletion"*, Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, pp. 259-270, Berlin, Germany, 2013.

[14] J. Reardon and H. Ritzdorf, *"Secure Data Deletion from Persistent Media"*,Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, pp. 271-284, Berlin, Germany, 2013.

[15] N. P. Karvelas and A. Kiayias, *"Efficient Proofs of Secure Erasure"*, Security and Cryptography for Networks, Springer Cham, pp. 520-537, Vol. 8642, 2014.

[16] M. Miao, J. Wang, J. Ma, and W. Susilo, *"Publicly verifiable databases with efficient insertion / deletion operations"*Journal of Computer and System Sciences archive, Vol. 86, Issue C, pp. 49-58, 2017.

[17] J. Reardon, D. Basin, and S. Capkun, *"SoK : Secure Data Deletion"*, 2013 IEEE Symposium on Security and Privacy, IEEE, pp. 1-15, Berkeley, CA, USA, 2013.

[18] J. Xiong et al., *"A secure data self-destructing scheme in cloud computing"*, IEEE Transactions on Cloud Computing, Vol. 2, Issue: 4, pp. 448-458, 2014.

[19] S. Ahmad and M. M. Afzal, *"A Review of Assured Data Deletion Mechanism in Cloud Computing"*, International Journal of Engineering & Technology, Vol. 7, pp. 329–332, 2018.

[20] H. Mu and Y. Li, *"An assured deletion scheme for encrypted data in Internet of Things"*, Advances in Mechanical Engineering, Vol. 11, No. 2, pp. 1–11, 2019.

[21] F. Shan et al., *"An Attribute-Based Assured Deletion Scheme in Cloud Computing"*, International Journal of Information Technology and Web Engineering (IJITWE), Vol.14, no. 2, pp. 74–91, 2019.

[22] A. Bentajer, M. Hedabou, K. Abouelmehdi, and S. El Fezazi, *"An IBE-based design forassured deletion in cloud storage"*, Journal Cryptologia, Taylor and Francis,Vol. 43, Issue. 3, pp. 1-12, 2019.

[23] D. Zhong, S. P. Liang, H. Xinfeng, D. U. Ruizhong, S. H. I. Pengliang, and H. E. Xinfeng, *"Cloud data assured deletion scheme based on overwrite verification"*,Journal on Communications, Vol. 40, Issue: 1,pp. 130-140, 2019.

[24] S. M. Diesburg and A. A. Wang, *"A Survey of Confidential Data Storage and Deletion Methods"*, ACM Computing Surveys (CSUR) Surveys, Vol. 43, Issue: 1, 2010.

[25] Y. Sun, J. Zhang, Y. Xiong, and G. Zhu, *"Data Security and Privacy in Cloud Computing"*, International Journal of Distributed Sensor Networks, Hindawi Publishing Corporation, Vol. 2014, Volume 2014, Article ID 190903, 2014.

[26] L. Xue, Y. Yu, Y. Li, M. H. Au, X. Du, and B. Yang, *"Efficient Attribute-based Encryptionwith Attribute Revocation for Assured Data Deletion"*,Information Sciences, Vol. 479, pp. 640-650, 2018.

[27] Yang, X. Chen, and Y. Xiang, *"Blockchain-Based Publicly Verifiable Deletion Scheme for Cloud Storage"*, Journal of Network and Computer Applications, Vol. 103, pp. 185-193, 2018.

Authors Profile

Ms. Seema Joshi is working as Assistant Professor (Cyber Security) at GTU-Graduate School of Engineering and Technology, Gujarat Technological University, Chandkheda, Ahmedabad, Gujarat. She has worked in many Government Projects. She has published many papers in journals and conferences and two books with Pearson Education.

Prof. (Dr.) Shaileshkumar D. Panchal is working as an Associate Professor (GES, Class-I) in Information Technology Department at Vishwakarma Government Engineering College, Chandkheda, Ahmedabad, Gujarat. He has 19 years of teaching experience. He has completed PhD in Computer Engineering from CHARUSAT in year 2017. He has published many research papers in referred journals.

