Privacy Preserving Data Access To Cloud

Preetha S, Vishwas Setty N, Siddharth S, Rakesh G N

Abstract: ***The current systems stress on protection of data stored in the cloud servers without giving much thought and consideration to the protection of data during user access. Encryption of data is a technique that is popularly used to protect stored data. Encryption essentially scrambles the data and stores it in a form which makes no sense unless decrypted with the suitable key. Every cloud service provider ensures data is stored in an encrypted form in its servers. Encryption of data is not sufficient to protect user data as acquiring the appropriate key can result in decrypting of the data. Encrypting the data before uploading the data to the cloud can help to an extent to preserve data. To access the data it would need to be encrypted twice- once by the cloud service provider and then by the user. Cloud service provider is prevented from accessing user data and also other third-party individuals. However, this approach too is not efficient and sufficient to protect user data. ORAM algorithm is used to enable access to user data stored on distributed file systems that comprises of multiple servers stored either at a single location or multiple locations across the globe in a manner which ensures the user privacy is protected when accessing the data. Reshuffle of data blocks stored in third party servers ensures the access pattern of the user remains hidden. ORAM algorithm does not cause any hindrance to the data access and does not lead to any major drop in data access rate. To ensure security, we propose a load balancing technique to guarantee smooth and safe approach for data access.***

Index Terms**: Cloud, Data Access, Dropbox, ORAM.**

I. INTRODUCTION

Data can be defined as various pieces of information, usually presented in a special way. Since many decades, people have used the word data in order to mean computer information that is transmitted or stored. An example of data is the information regarding a business stored in a database or some storage media.

Data storage can be referred to the concept of storing or preserving data on some piece of hardware or device for future access. Storage is referred to both a user's data generally as well as the integrated hardware and software systems that use data for operations or analysis. Storage is of utmost importance in today's world where it is extensively used to store personal data as well business-related content. This includes information in applications, databases, data centres, archives and cloud storage. The storage facilities are expanding in terms of size and also speed.

Cloud storage is a service offered by third party vendors who make use of multiple servers to store user data at a specific price. The users (usually businesses or organizations) buy the storage service from the providers to store business, application or other data. Cloud storage is a form of virtual environment comprising of multiple servers which allow for storage of large amount of data from multiple sources belonging to numerous users. One of the major differences between traditional storage and cloud storage is with respect to how data is accessed. Most cloud service providers make use of web-based applications to allow users access to their data while other less popular approaches too are in use. Hence, majority of the API calls made by the cloud services are based upon the HTTP protocol. Many cloud service providers make use of this, some of them being Microsoft Azure, Mezeo Cloud Service and Amazon S3.HTTP based protocols alone wouldn't be sufficient to access the data from the cloud, there arises the need to integrate the HTTP protocol with some other applications such as NFS/Common Internet File System, FTP or iSCSI that allow data retrieval and storage.

"Big data has emerged in various domains including science, engineering and commerce. The access privacy problem was first addressed by private information retrieval (PIR) technique that allows a user to retrieve a block from a database of N items held by a server that learns nothing about this block. Oblivious RAM (ORAM) is proposed to conceal information get to security with improved execution. The fundamental thought is to occasionally reshuffle information squares put away in an untrusted server with the end goal that client get to can't be followed. This paper applies the ORAM calculation to empower security saving access to huge information in mists."

II. LITERATURE SURVEY

Managing the test of pleasing immense volume of information that constantly develops in high speed, huge information is put away in dispersed record frameworks based upon hundreds or thousands of servers in a solitary or different geo-conveyed cloud locales. ORAM is straight forwardly connected on such disseminated stockpiling frameworks, regardless of whether all information squares are uniformly gotten to by clients, get to stack on servers would be truly lopsided [1]. Few models that are similar to Oblivious RAM do not leak any information to the attacker, but they are very expensive to be practical on large datasets. Other symmetric key encryption-based models propose effective answers for the issue, yet spills information to design. In such pursuit plans, given a question x of watchword w, an aggressor does not learn w, but realizes which are the records that contains it. In [2], Islam et.al distinguishes the potential dangers of access design by means of a novel approach. The exact examination with a true email dataset demonstrates that huge measure of sensitive data that can be undermined by the proposed assault.

Revised Manuscript Received on July 10, 2019.

Preetha S, Department of ISE, BMSCE, Bangalore, India.
Vishwas Setty N, Department of ISE, BMSCE, Bangalore, India.
Siddharth S, Department of ISE, BMSCE, Bangalore, India.
Rakesh G N, Department of ISE, BMSCE, Bangalore, India.

Data masking is one of the methods for creating a structurally similar but non authentic version of an organization's data which can be used for purposes like software testing and user training. The purpose is to secure the real information while having a functional substitute for events when the genuine information is not required. Albeit most associations have stringent security controls set up to ensure information creation in business use. Equivalent information has been utilized for activities that are less secure. In [3] information masking, organization of information are done by changing the qualities. Information is adjusted in various ways including encryption, character rearranging, and character or word substitution. Security issues for huge numbers of frameworks in a cloud must be secured. Numerous security issues for cloud computing encompasses many technologies including networks, virtualization, load balancing, concurrency control, database, operating system, memory management are discussed in [4].

HAIL (High Availability and Integrity Layer) is a system with distributed cryptography where a number of servers ensure a client that, the stored files are recoverable and not harmed[5]. HAIL systems are highly compact and proofs can be efficiently computed irrespective of the file size. HAIL proves and actively allots file shares. Proofs of Retrievability (POR) implemented on individual servers is made more efficient and secured by HAIL. A new method for privacy protecting access control scheme is assessed for data security. The real and genuine identity of a user is verified by the cloud without knowing user's identity. An aspect of access control is allowed only to users with privilege are able to decrypt the stored information.

Cyptographic-based access control is implemented in order to support privacy-preserving authentication and authorization for data outsourcing scenario. In [6] Fugkeaw et.al proposes a novel access control model consolidating Role-based Access Control (RBAC) model, Symmetric encryption, and Ciphertext quality-based encryption (CP-ABE) to help fine-grained get to control for huge information re-appropriated in distributed storage frameworks. Ciphertext is related with a lot of traits for all with an open key segment is characterization. Client mystery key is developed to connect with an entrance structure. In any case, the ABE plans come up short on the specialist authority over the arrangement for implementation. Ciphertext Policy Attribute Based Encryption (CP-ABE) is utilized to address this weakness. Fugkeaw, examined whether by imitating the database, progressively productive answers for the private recovery issue can be acquired - a plan that empowers a client to get to 'k' repeated duplicates of a database and secretly recover data put away in the database. This implies every individual server holding a reproduced duplicate of the database gets no data on the character of the thing recovered by the client. Client who has some record i, and is keen on acquiring the estimation of the bit xi, questions all from the servers and acquires answers from which the ideal piece xi can be registered. The question to every server is disseminated autonomously of i and this way every server picks up no data about i. A plan with these properties is known as a Private Data Recovery (PIR) conspire. Chor et al[7] presented different PIR plans with completely littler correspondence multifaceted nature than n-bit arrangement. Combining Private Information Retrieval (PIR) techniques with the likely bandwidth efficient existing ORAM scheme known as ObliviStore was proposed in [8]. ObliviStore and Private Information Retrieval both created a new ORAM with bandwidth costing only half those of ObliviStore by eliminating waste dummy blocks from the tree storage structure used by the ORAM algorithm.

Path ORAM an incredibly basic Oblivious RAM with a little measure of client storage likewise having an O(logN)bandwidth cost was presented. Path RAM is the most feasible ORAM plot known till date with little customer stockpiling and straightforwardness [9]. ORAM is a cryptographic method handles data spillage in a program's memory. The fundamental thought is that ORAM keeps up a scrambled and rearranged structure for all information put away in memory. For every memory to reach, information is encrypted and reshuffled. In ORAM, a memory gets to design is computationally unclear from the others with a similar length. In [10] the method utilizes a condition of-craftsmanship ORAM plan to be specific, Way ORAM for instance to represent the memory get to stream after an LLC miss in ORAM. Way ORAM is a proficient and algorithmically basic Neglectful Slam, which has been proposed to be a segment in a protected processor. Late models have demonstrated that Way ORAM is a standout amongst the most encouraging and productive ORAM usage. Oblivious RAM (ORAM) allows a client to read and write data hosted by an untrusted cloud party; it is also responsible for hiding both the data and the access pattern from the untrusted host. Access pattern privacy is a critical component of data privacy. Without access design protection, the demonstration of perusing and composing remote information spills possibly basic data about the information itself, making it difficult to accomplish full information classification. The essential thought behind SR-ORAM is to crease the intuitive questions into a solitary non-intelligent solicitation without yielding security. Blossom channel ORAM is used in [11] as it allows itself advantageously to utilization of a non-intelligent inquiry object and gives protections against effectively noxious enemies. Randomized shell sort characterization permits more stringent customer stockpiling prerequisites of SR-ORAM.

A cloud-oriented authenticated file system called Iris which gives clients an comprehensive, efficient and real-time data-integrity verification was proposed in [12]. The Iris system enables an enterprise client or an auditor acting on the client's behalf to verify the integrity and genuineness of any data retrieved from the file system while performing typical file system operations. Iris offers a design versatile to numerous customers (on the request of hundreds or even thousands) issuing activities on the file framework in parallel. Iris incorporates new advancement and endeavour side reserving systems specifically intended to conquer the high system dormancy normally experienced while getting to distributed storage. Iris also incorporates novel eradication coding procedures for efficient backing of dynamic Verifications of Retrievability (PoR) conventions over the file framework.

III. PROPOSED MODEL

ORAM algorithm is proposed to overcome the issue of data privacy during access from the cloud.

The algorithm ensures the access pattern of the user which cannot be traced by the cloud service provider or any other individual other than the owner of the data. The algorithm shifts the user data between multiple nodes of the cloud storage structure before finally storing it in a particular node the location, which is known only to the user.

In the era of cloud computing, project deployment is done on AWS. A full-fledged ORAM application that is attached to a load balancer is set up on AWS. The service is hosted on the on-Apache Tomcat 8. The Tomcat is not customized; it is deployed as is when installed onto EC2. All customizations will occur post installation. Tomcat has a dependency on JDK 8 therefore JDK is installed from S3 bucket. The application has a health check page which signifies the health of the application. EC2 instance is safe & secure from any unwanted incoming or outgoing traffic. The mod_wsgi Apache module is used for serving Python scripts over HTTP via Apache web server and WSGI (Web Server Gateway Interface) is an interface between web servers and web apps for python and also as ORAM creates load balancing issues an elastic load balancer has been set up to achieve load balanced storage system with improved availability and responsiveness.

File Encryption with AES

AES being a symmetric algorithm, uses the same key at the sender and the receiver's end for encrypting and decrypting respectively. Depending on the block size of data being encrypted and the size of the cryptographic keys used. The number of rounds of encryption process varies. 128-bit keys posses 10 rounds, for 192-bit keys -12 rounds and 256-bit keys - 14 rounds. Each round consists the following steps – Byte Substitution, ShiftRows, MixColumns and AddRoundKey.

CBC-MODE with IV (128-bit)

AES encryption uses a technique called Cipher block chaining (CBC). As AES encrypts on blocks of data of a particular size it is essential to make sure data is of the required size. Data size is ensured using CBC, which allows encryption of sequence bits as a single unit or with a cipher key applied to a block. CBC mode of AES encryption requires an Initialization Vector (IV) for each round of operation. During decryption the data, the same IV has to be used which was used in encrypting the data in that particular round.

Padding with PKCS7

AES encrypts on blocks of data of a particular size it is essential to make sure data is of the required size. PKCS7 padding is used to ensure the required size of data. PKCS7 padding appends additional bytes to a pre-existing block of data to make the data block of suitable size for encryption. Padding generally refers to addition of extra data at the beginning, middle or end of pre-existing data.

In PKCS7, padding is done in whole bytes. The padding byte that is added, represents the number of bytes being added. For example a padding of 01 represents that 1 byte of data is added. The number of bytes added will depend on the block boundary to which the message needs to be extended.

HMAC using SHA256

HMAC (hash-based message authentication code) uses cryptographic hash function (SHA-256 or SHA-3) and a cryptographic key resulting in the generation of a message authentication code (MAC).This may be used to verify the authenticity of any message as done by any MAC. The HMAC algorithm is termed HMAC-N, where N is the cryptographic hash function used i.e. SHA256 or SHA3. The effectiveness of a HMAC is based upon the size of the cryptographic has function and cryptographic key size.

AMAZON EC2

Ec2 instance acts like a remote server. You can deploy your application onto the ec2 instance and the amazon cloud will take care to host it. One can also choose to select the required operating system but the operating system chosen sometimes might not be free. The free trier operating systems that are available are free of cost but there will be costs incurred because of hosting an ec2 instance. You don't have to explicitly buy the hardware to deploy your application. This feature can be used to host any number of virtual servers based on the need.

Amazon EC2 Auto Scaling

Auto scaling group increases or decreases the number of Ec2 instances when the load rises or shrinks. It is used to group the number of Ec2 modules. Each auto scaling group belongs to some particular region or the subnets. AWS provides various auto scaling group policies that used. Defining our own custom policy for the auto scaling group can be done. It also provides health check facilities like health check grace period etc. The main feature of this group is to automate the number of EC2 instances at any given time.

Oblivious RAM

Oblivious RAM (ORAM) uses untrusted RAM (i.e. the cloud storage) through a trusted processor. ORAM is represented as large blocks called caches with each cache comprising of numerous buckets as shown in figure 1. These blocks form a tree like structure which is stored in the cloud storage where the data are added to it. When a user makes a request for a data block, the algorithm looks for another block known as the dummy block which serves as the temporary block where the data gets stored to prevent tracing the user's access path. This way the algorithm shifts the data block upwards till it reaches the top most block from where the user collects the data. As each level of blocks starts getting filled, data gets shifted downwards eventually getting stored in a block the location of which is known only to the user.

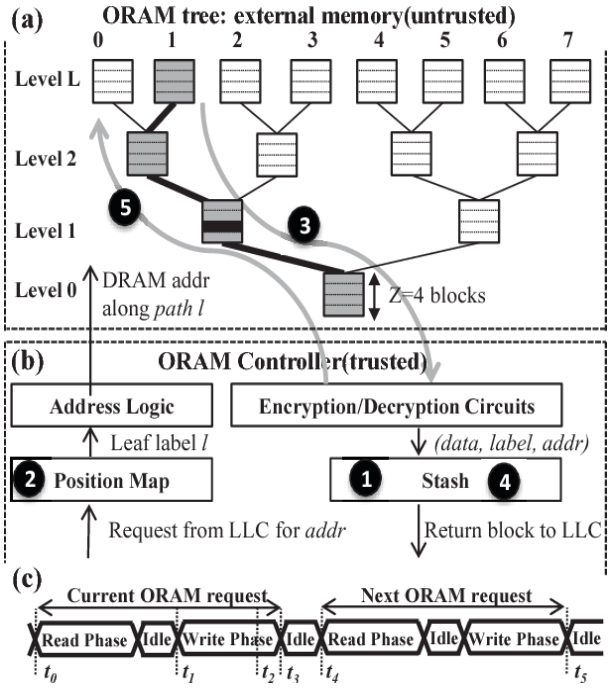

Fig 1: Oblivious RAM representation

A data access sequence is represented as A = ⟨ (oper1, addr1, d1), (oper2, addr2, d2) . . . ⟩ ,
where, oper-i represents read or write, addr-i is block address and d-i represents data.
Data is usually stored in the form of a tree like structure on the cloud where each node represents a block consisting of numerous buckets. ORAM makes use of two basic operations

- Read And Remove and Add.
- Read And Remove(u): This operation reads data from a block
  with address u and removes it from that block.
- Add(u, d): This operation writes data d into a block having address u.

To perform read multiple ReadAndRemove(u) followed by an Add(u, d) operations is performed. To perform write redundant ReadAndRemove(u) before Add(u, d) is performed. The multiple read and write operations allow the access pattern of the user to be hidden.
When a block of data is added to the cloud storage it is added to the top most block at Tier 0 eventually, as more data is added the top most block starts getting filled resulting in data to be pushed down to the lower blocks. Finally, the user is the only who knows in which block the data get stored. To retrieve a stored block of data, the user passes the address of the block where the data is stored to the algorithm. The algorithm then shifts the data from that block to another block till it reaches the outermost block. The use of the dummy blocks to shift the data prevents the cloud and other unauthorized individuals from getting to know the user's access pattern.

IV. EXPERIMENT AND RESULTS

A. Uploading a file to random node

The proposed method allows an option for the client in order to upload a given file to the data storage medium as shown in figure 2, we have considered Dropbox. The file should be stored in random nodes so that we can preserve the data access to that particular file. ORAM algorithm is designed to work in that way.

Fig 2: Uploading a file to a random node

Fetching data from stored node
Client is allowed to download or fetch a given file from the data storage medium i.e. from Dropbox. The data that is stored in random nodes is only accessible to the client who is the owner of that particular data. Figure 3 represents the data fetched from the stored node. When data is fetched from the storage medium it is provided with proper security and encryption for that file and privacy is preserved while accessing the data.

Fig 3: Fetching data from the stored node

Failed Scenario
Uploading the file requires to initially store it on the device which fails in case the memory allocated is not sufficient and has to be changed manually through the code.

Integration Testing
Upon creation of various EC2 instances in the AWS framework and implementing the ORAM algorithm we need to again test for reliability and accessibility.

Failed Scenario
Setting up of the EC2 instances efficiently is a difficult process and in some cases my fail to divert the traffic to servers efficiently as shown in figure 4.

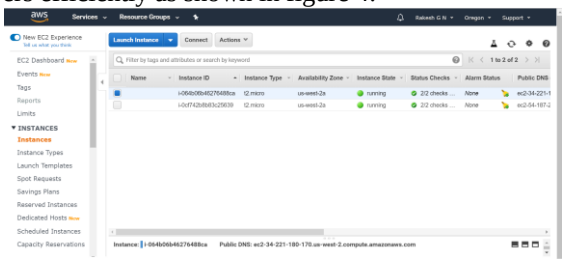

Fig 4 : Setting up EC2 instances

System Testing
Figure 5 shows the system unit of testing process.

After integration testing of every module, it is necessary that we need to implement the complete package in a single system. After system testing, one can guarantee about the reliability and security.

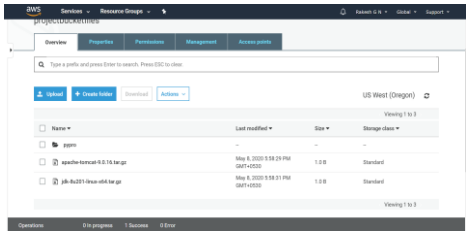

Fig 5: System unit

Key Findings

Cloud service providers have their own mechanisms to ensure protection of user data. The application of the ORAM algorithm to scenarios is very useful in complete protection of user data. ORAM algorithm periodically reshuffles blocks of data shifting them from one node of the cloud storage structure to another thereby making it harder for unauthorized individuals as well as the cloud service providers from tracing the access pattern of the user. With the user of the data being the only one who knows where it is stored, the data remains completely protected.

Applications of ORAM algorithm is observed to have a phenomenal of unbalanced load across multiple servers. To handle such scenarios, we have implemented the same using AWS EC2 instances which help in efficient distribution of the load across the servers. Uploading and downloading a file is shown in figure 6 and 7.

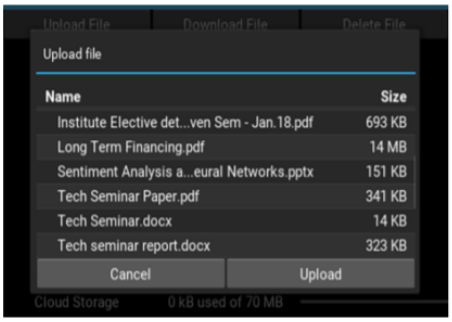

Fig 6: Uploading a file

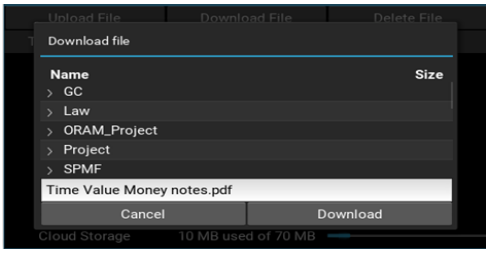

Fig 7: Downloading a file

The node structures in dropbox can be viewed as shown in figure 8.

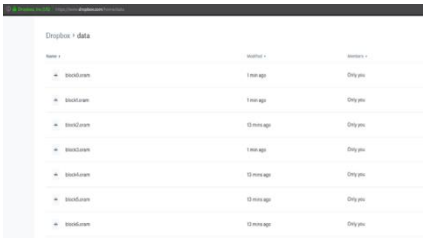

Fig 8: Nodes on Dropbox

Limitations

Since the cloud services operate using hundreds and thousands of servers located at different parts of the globe, the application of the ORAM algorithm results in issues resulting in load balancing. Some of the servers may have more data requests forwarded to them while others may be free. To handle such problem the ORAM algorithm was implemented on various EC2 instances provided by AWS. Balanced load across all the servers can be achieved. However, the use of the AWS services requires the payment of a particular charge.

Difficulties Encountered

Understanding of the concepts regarding the working of the ORAM algorithm was a challenging one as it required knowledge regarding various types of data access methods that preserve privacy. Since, there are various types of ORAM algorithms - Path-ORAM, S-ORAM, SCR-ORAM etc., the selection of the most ideal one which was ideal for this situation was also to be taken care of.

Also, dealing with issue of load balancing which required setting up a load balancer in AWS and setting up EC2 instances for the same was a challenge.

V. CONCLUSION

Our approach deals issues related to data privacy when accessing the data from cloud servers. We have proposed the use of the ORAM algorithm in association with encryption and other data protection techniques. The proposed ORAM algorithm was initially applied to local data storage which was later applied to a cloud service provided by Dropbox. However, using only the ORAM algorithm led to the issue of unbalanced load across the multiple servers, which led to the use of a data placement problem to achieve load balance. Hence, we have implemented the same in AWS EC2 with the help of Load balancer, Auto scaling groups and Elastic load balancers.

We observed the absence of load balancing issues on application of ORAM.

ACKNOWLEDGMENT

The authors would like to acknowledge BMS college of Engineering and TEQIP III phase for their immense support in carrying out and encouraging this research work.

REFERENCES

1. Li, Peng, and Song Guo. "Load balancing for privacy-preserving access to big data in cloud." 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2014.
2. Islam, Mohammad Saiful, Mehmet Kuzu, and Murat Kantarcioglu. "Access Pattern disclosure on Searchable Encryption: Ramification, Attack and Mitigation." Ndss. Vol. 20. 2012.

3. Garg, Sushil. "Survey on Cloud Computing and Data Masking Techniques." Accessed December 15 (2017): 2017.
4. Hamlen, Kevin W., and Bhavani Thuraisingham. "Data security services, solutions and standards for outsourcing." Computer Standards & Interfaces 35.1 (2013): 1-5.
5. "Li, Peng, and Song Guo. "Load balancing for privacy-preserving access to big data in cloud." 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2014.
6. Fugkeaw, Somchart, and Hiroyuki Sato. "Privacy-preserving access control model for big data cloud." 2015 International Computer Science and Engineering Conference (ICSEC). IEEE, 2015.
7. Chor, Benny, et al. "Private information retrieval." Proceedings of IEEE 36th Annual Foundations of Computer Science. IEEE, 1995.
8. Dautrich, Jonathan, and Chinya Ravishankar. "Combining ORAM with PIR to minimize bandwidth costs." Proceedings of the 5th ACM Conference on Data and Application Security and Privacy. ACM, 2015.
9. Stefanov, Emil, et al. "Path ORAM: an extremely simple oblivious RAM protocol." Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. ACM, 2013.
10. Zhang, Xian, et al. "Fork path: improving efficiency of oram by removing redundant memory accesses." Proceedings of the 48th International Symposium on Microarchitecture. ACM, 2015.
11. Williams, Peter, and Radu Sion. "Single round access privacy on outsourced storage." Proceedings of the 2012 ACM conference on Computer and communications security. ACM, 2012.
12. Stefanov, Emil, et al. "Iris: A scalable cloud file system with efficient integrity checks." Proceedings of the 28th Annual Computer Security Applications Conference. ACM, 2012.
13. Pawar, Chandrashekhar S., Pankaj R. Patil, and Sujitkumar V. Chaudhari. "Providing security and integrity for data stored in cloud storage." International Conference on Information Communication and Embedded Systems (ICICES2014). IEEE, 2014.
14. Gahi, Youssef, Mouhcine Guennoun, and Hussein T. Mouftah. "Big data analytics: Security and privacy challenges." 2016 IEEE Symposium on Computers and Communication (ISCC). IEEE, 2016.
15. Shi, Elaine, et al. "Oblivious RAM with O ((logN) 3) worst-case cost." International Conference on The Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2011.
16. Garg, Sushil. "Survey on Cloud Computing and Data Masking Techniques." Accessed December 15 (2017): 2017.
17. Liu, Chang, et al. "Oblivm: A programming framework for secure computation." 2015 IEEE Symposium on Security and Privacy. IEEE, 2015.
18. [18] Ostrovsky, Rafail. "Efficient computation on oblivious RAMs."Proceedings of the twenty-second annual ACM symposium on Theory of computing. ACM, 1990.
19. Wang, Xiao, Hubert Chan, and Elaine Shi. "Circuit oram: On tightness of the goldreich-ostrovsky lower bound." Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, 2015.
20. Fletcher, Christopher W., et al. "Freecursive oram:[nearly] free recursion and integrity verification for position-based oblivious ram." ACM SIGARCH Computer Architecture News 43.1 (2015): 103-116.
21. "Priya Dhir Research Scholar (RIMT UNIVERSITY). "Survey on Cloud Computing and Data Masking Techniques""
22. "Jeet Vyas, Prof: Prashant Modi Computer Engineering, Ganpat University, Information Technology, Ganpat University. "Providing Confidentiality and Integrity on Data Stored in Cloud Storage by Hash and Meta-data Approach""
23. "International Journal of Innovations & Advancement in Computer Science IJIACS, ISSN 2347 –" Survey on Cloud Computing and Data Masking Techniques" 8616,Volume 6, Issue 4,April 2017"

AUTHORS PROFILE

Preetha S, Assistant Professor, working in BMS College of Engineering, affiliated to VTU, Bangalore, India. She is a research scholar and a full time faculty member in the Department of Information Science & Engineering. She completed her Master's Degree in Computer Network Engineering from Visvesvaraya Technological University. Her research area includes Biometrics and Wireless Sensor Networks.

Vishwas Setty N, student of Information Science and engineering, BMS College of Engineering, Bangalore. His interests include cloud computing.

Siddharth S, student of Information Science and engineering, BMS College of Engineering, Bangalore. His interests include cloud computing.

Rakesh G N, student of Information Science and engineering, BMS College of Engineering, Bangalore. His interests include cloud computing.