



BIFOLD CONNECTION MANAGEMENT FOR AWS CLOUD-BASED INFORMATION PROCESSING AND DISTRIBUTION

A. Vijay Kumar¹ Mr. N. Muni Sankar²

¹ Department of CSE, Sri Venkatesa Perumal College of Engineering and Technology, Tirupati, India, vijayanukum@gmail.com

² Associate Professor & HOD, Sri Venkatesa Perumal College of Engineering and Technology, Tirupati, India ms.nagugari@gmail.com

Abstract— Recent years have seen a rise in interest in AWS Cloud-based data storage services, both from academia and business, due to their efficient and low-cost administration. Service providers must utilize secure data storage and sharing mechanisms to protect data confidentiality and service user privacy since they deliver services in an open network. When it comes to protecting sensitive data, encryption is the most frequently utilized approach. In the real world of data management and administration of the data, encryption (e.g., AES) is not enough. To prevent Economic Denial of Service (EDoS) attacks, an effective access control over download requests should also be considered. Using AWS AWS Cloud-based storage as an example, we build a control system for both data access and download requests that doesn't compromise security or efficiency. These two bifold access control systems are each created for a specific design scenario in this article. In addition, the systems' security and experimental analyses are discussed.

Keywords:-Searchable Encryption, Multi-Keyword Search, Multi-User Access, Search Pattern, Access Pattern

I. INTRODUCTION

Over the past few decades, AWS Cloud-based storage services have drawn the interest of both academics and industry. As a result of its broad list of benefits, including access flexibility and free local data management, it may be widely utilized in various Internet-based commercial applications (e.g., Apple I Could). Individuals and businesses are increasingly turning to the AWS Cloud to store and manage their data in order to avoid the expense of updating their local data management facilities and devices. Internet

consumers may be deterred from adopting AWS Cloud-based storage services because of concerns about security breaches. There are a number of situations in which outsourced data may need to be shared with others in order to be used effectively. If you're a Dropbox user named Alice, you might be able to exchange photographs with your pals via the Dropbox application. It is necessary for Alice to establish a sharing link and then share it with her friends in order to share photographs without data encryption. Even if the sharing link is hidden from unauthorized users (e.g., those who aren't Alice's friends), it is visible at the Dropbox management level (e.g., administrator could reach the link). To protect data security and privacy, it is typically advised to encrypt data before uploading it to the AWS Cloud.

In this case, one option is to encrypt the data before uploading it to the AWS Cloud, such that only a specific AWS Cloud user (with a valid decryption key) may decode the data. Encrypting the material before sharing it with others is an easy approach to prevent "insiders" from seeing shared photographs. It's possible that Alice has no idea who the photo recipients/users will be. Alice may only be aware of the properties of picture receivers, which is feasible. Because the encryption must know in advance who the data receiver is, standard public key encryption (e.g. Paillier encryption) is not an option here. To ensure that only authorized individuals may view the photographs, Alice should have access to a policy-based encryption mechanism over the outsourced photos.

Known as a resource-exhaustion attack, resource-exhaustion attacks are frequent in AWS Cloud-based storage services. A malicious service user may launch denial-of-service (DoS)/distributed denial-of-service (DDoS) attacks on a AWS Cloud storage service server to consume the server's resources so that the AWS Cloud service is unable to respond to honest users' service. Since a public AWS Cloud may not have



any control over download requests (namely, a service user may send unlimited numbers of download requests to AWS Cloud server), Because of this, economic components of the "pay as you go" model might be affected owing to increased resource use. Users of AWS Cloud services will see their bills skyrocket.

As a solution to these two issues, we suggest in this work a novel method called dual access control. It's possible that attribute-based encryption (ABE) [9] might be a good option for securing data in a AWS Cloud-based storage service. ABE allows for the confidentiality of outsourced data as well as fine-grained management of the outsourced data. There are a number of data encryption methods available, including Ciphertext Policy ABE (CP-ABE) [5]. It should be noted that this article considers the usage of CP-ABE as part of our methodology. Although CP-ABE may be used to create a sophisticated system that ensures the control of both data access and download requests, it is not sufficient.

II. RELATED WORKS

To prevent key-scraping attacks, [15] presents a hybrid encryption scheme that may be revoked. [16] The key idea of the study is AONT [2]. A method with optimum asymmetric encryption padding was chosen by us. since reversing it required knowing the full output. In this way, reversing OAEP becomes impossible by altering random bits. Since bits are altered throughout the encryption process, it is necessary to store them so that they may be restored afterwards.

However, this means that the size of the cipher text increases with each re-encryption. Decrypting multiple times-encrypted files is therefore a time-consuming process. They also suggest that AONT might be implemented by servers in order to make this method more efficient. A completely trustworthy server is required for this, thus internal assaults are not prevented. According to [5], functional encryption is used to create a protocol, with the major functionality executing in separate contexts.

In SGX enclaves, a file is decrypted and a function f is applied to the decrypted file. More than that, all enclaves may verify each other and exchange data through secure communication channels. " In our design, we combine SSE and ABE

to create a hybrid encryption method, despite the fact that we employ the same hardware principles. Defensive elements of the protocol are also executed and sensitive data is stored using SGX. It is described in [9] as an attribute-based encryption method based on cipher text policy attributes. It's possible to revoke an encryption key by attaching it to the cipher texts. Users' keys will expire after a specific amount of time in order to prevent keeping extensive revocation lists.

So, the revoked key list only includes revoked keys that have not yet expired. The authors of [6] suggest a hybrid encryption method that uses SSE and ABE. Users encrypt their files using SSE in the proposed system, but the indexes that arise are encrypted using ABE. Users' search tokens can then be produced locally and transmitted to the cloud. In order for search to work, the user's attributes must meet the encrypted index's policy. A static approach may seem like a good idea but it can only be utilized in a limited number of real-life situations. A revocation mechanism is also missing, which is crucial for cloud-based applications [11, 18].

Using the SGX feature and separating it from the ABE scheme, we have designed an efficient revocation method that overcomes these difficulties.

III. METHODOLOGY

Data Transferring is the major task to secure data we use encryption techniques to hide the content, in this the study examines the importance of information to protect information, and encryption techniques are used. In this web application there is some modules called CSP (Cloud Service Provider), Authority, Data owner, and Data User. The Core concept is Bifold Access in between data owners and data user with in the access of CSP and Authority. CSP plays a major role to access data owners request then data owners can view the all the data owners who are accessed by the Cloud service provider and he can view the request from the users through authority and sends key to the user.

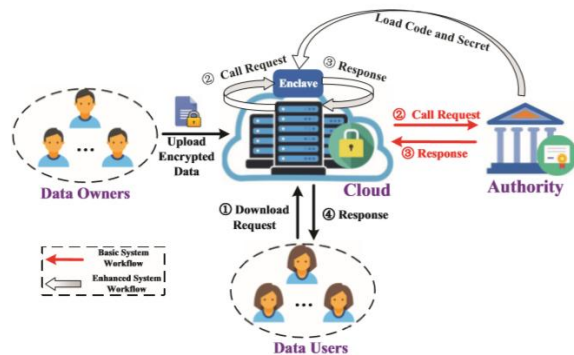
Data owners plays a vital role in uploading files into AWS cloud and files are stored by providing keyword for each file and encrypted then view files in encryption format then he can view all the data owners files which are in encrypted format the data owner has to send request to data owner for such file if accepts then only data can be viewed in text format .Authority is the module which receives request from the data user for files and

authority passes that request to Clod Service Provider the CSP generates key and sent to the user. Data user login into the system and search files using Keywords then sends request to the Authority. Authority then sends that request to CSP and CSP generates key and sent to user finally user can view information using that key only.

Data owner download that file based on his requirement. Everything is stored in S3 buckets on Amazon Web Services (AWS). Amazon S3 Bucket allows data owners to download data files. Buckets are the containers for objects. The buckets can be one or more and can be placed in whatever order you like. This means you can pick who has permission to add and remove things from the bucket and check access logs for them. An object-storage service, Amazon Simple Storage Service (Amazon S3), is one of the most popular cloud storage services. Amazon S3 allows you to store and access any amount of data from anywhere.

Using Amazon Relational Database Service (Amazon RDS), you can quickly and easily set up a relational database in the cloud and scale it up or down as needed. There's no limit to the amount of data you may store, and you can scale up or down according to your needs. As a result, you're able to focus on your applications and provide them with the speed, availability, security, and compatibility that they require with no effort.

You may use Amazon RDS with a variety of database instance kinds - memory, performance, or I/O-optimized - and select from six well-known database engines, such as Amazon Aurora, PostgreSQL. Your current databases may be simply migrated or replicated to Amazon RDS using the AWS Database Migration Service.



IV. RESULTS

Figure: 1

Cloud Service Provide

Cloud service provider having default login credentials with those he log in into the system and perform his operations.

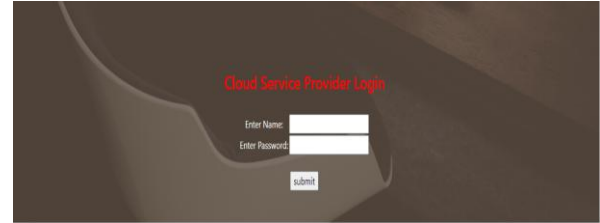


Figure: 2
View Owners

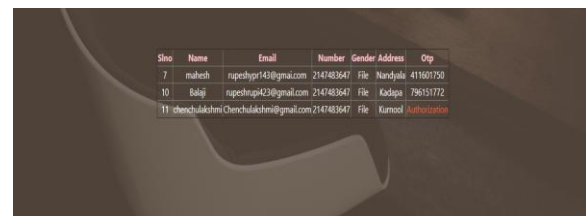
After login CSP can view data owners and send otp to the data owners, view users, and user's request.



Sino	Name	Email	Number	Gender	Address	Otp
16	Rupesh	rupeshr@gmail.com	2147483647	Male	Piles, chittoor(dst), AP	2343256
17	Sainath	Sainath@gmail.com	2147483647	Male	puttur(dst)	69878
19	Narendra	Narendra@gmail.com	2147483647	Male	Kadapa	54408
20	Jayaram	Jayaram@gmail.com	2147483647	Male	tinupati	276943
21	Hari	Hari@gmail.com	2147483647	Male	Banglore	Authorization

Figure: 3
Authority

Authority is also having default credentials with those credentials only authority can login into the system and perform operations like view users request and key Generation and view users for authorization.



Sino	Name	Email	Number	Gender	Address	Otp
7	mallesh	rupeshr14@gmail.com	2147483647	File	Nandyala	411601758
10	Babji	rupeshr142@gmail.com	2147483647	File	Kadapa	796351772
11	Chenchukochi	Chenchukochi@gmail.com	2147483647	File	Kurnool	Authorization

Figure: 4
Data Owner

Data Owner is the person who login into the system after registration, Data Owner Perform operations like upload files and view files (encrypted view), view all files, send request for dual access and view dual request response.



Figure: 5
Data User

Data user registers into the System the logs into the system then data user can view his profile and search for file with the help of keywords then send's request to the Authority and Authority passes that request to the Cloud, if cloud accepts then key will be generated to the user

V. CONCLUSION

In this article. Cloud-based data sharing presents a fascinating and long-standing challenge that we solved by presenting two dual access control methods. DDoS/EDoS assaults are not a problem for the suggested solutions. "Transplantable" CP-ABE structures are described as a result of the approach utilized. No significant computational or communication overhead was found in our experiments (compared to its underlying CP-ABE building block). Enclaves are used to protect secret information from being accessed, and our system takes use of this feature. Enclave may disclose part of its secrets to a hostile host through memory access patterns or other side-channel assaults, according to new research. Transparent Enclave Execution Model is introduced as a result of this for cloud data sharing, the challenge is to create an enclave with two levels of access restriction. As part of our future study, we'll examine the problem's related solution and will host the application in AWS cloud.

[1] Ittai Anati, Shay Gueron, Simon Johnson, and Vincent Scarlata. Innovative technology for cpu based attestation and sealing. In Workshop on hardware and architectural support for security and privacy (HASP), volume 13, page 7. ACM New York, NY, USA, 2013.

[2] John Bethencourt, Amit Sahai, and Brent Waters. Cipher text-policy attribute-based encryption. In S&P 2007, pages 321–334. IEEE, 2007.

[3] Victor Costan and Srinivas Devadas. Intel sgx explained. IACR Cryptology ePrint Archive, 2016(086):1–118, 2016.

[4] Ben Fisch, Dhinakaran Vinayagamurthy, Dan Boneh, and Sergey Gorbunov. IRON: functional encryption using intel SGX. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, pages 765–782, 2017.

[5] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Advances in Cryptology-CRYPTO 1999, pages 537–554. Springer, 1999.

[6] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In ACM CCS 2006, pages 89–98. ACM, 2006.

[7] Jinguang Han, Willy Susilo, Yi Mu, Jianying Zhou, and Man Ho Allen Au. Improving privacy and security in decentralized ciphertext-policy attribute-based encryption. IEEE transactions on information forensics and security, 10(3):665–678, 2015.

[8] Christofer Hoff. Cloud computing security: From ddos (distributed denial of service) to edos (economic denial of sustainability). <http://www.rationalsurvivability.com/blog/?p=66>.

[9] Joseph Idziorek, Mark Tannian, and Doug Jacobson. Attribution of fraudulent resource consumption in the cloud. In IEEE CLOUD 2012, pages 99–106. IEEE, 2012.