JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 10, ISSUE 03, 2023

A Cost-effective Data Compression-based Unified Access Control for Encrypted Cloud Storage

Reethika Kakani[1], Sri Chandra Nihitha. V[1], Yukta Chimpidi[1], Dr. S. Pradeep[2]

[1]UG Student, Department of Computer Science and Engineering, Malla Reddy Engineering College for Women (UGC-Autonomous), Maisammaguda, Secunderabad, Telangana, India

[2]Asssociate Professor & Head, Department of CSE-IoT, Malla Reddy Engineering College for Women (UGC-Autonomous), Maisammaguda, Secunderabad, Telangana, India

ABSTRACT

People endorse the great power of cloud computing but cannot fully trust the cloud providers to host privacy-sensitive data, due to the absence of user-to-cloud controllability. To ensure confidentiality, data owners outsource encrypted data instead of plaintexts. To share the encrypted files with other users, Ciphertext-Policy Attribute-based Encryption (CP-ABE) can be utilized to conduct fine-grained and owner-centric access control. But this does not sufficiently become secure against other attacks. Many previous schemes did not grant the cloud provides the capability to verify whether a downloader can decrypt. Therefore, these files should be available to everyone accessible to the cloud storage. A malicious attacker can download thousands of files to launch Economic Denial of Sustainability (EDoS) attacks, which will largely consume the cloud resource. The payer of the cloud service bears the expense. Besides, the cloud provider serves both as the accountant and the payee of resource consumption fee, lacking the transparency to data owners. These concerns should be resolved in real-world public cloud storage. In this work, we propose a solution to secure encrypted cloud storages from EDoS attacks and provide resource consumption accountability. It uses CP-ABE schemes in a black-box manner and complies with arbitrary access policy of CP-ABE. We present two protocols for different settings, followed by performance and security analysis.

Keywords: Cloud storage, combining data owner side, cloud side access, Economic Denial of Sustainability.

1. INTRODUCTION

Cloud storage has many benefits, such as always-online, pay-as-you-go, and cheap. During these years, more data are outsourced to public cloud for persistent storage, including personal and business documents. It brings a security concern to data owners: the public cloud is not trusted, and the outsourced data should not be leaked to the cloud provider without the permission from data owners. Many storage systems use server-dominated access control, like password-based and certificate-based authentication. They overly trust the cloud provider to protect their sensitive data. The cloud providers and their employees can read any document regardless of data owners' access policy. Besides, the cloud provider can exaggerate the resource consumption of the file storage and charge the payers more without providing verifiable records, since we lack a system for verifiable computation of the resource usage.

Relying on the existing server-dominated access control is not secure. Data owners who store files on cloud servers still want to control the access on their own hands and keep the data confidential against the cloud provider and malicious users.

1.1 Encryption is not sufficient

To add the confidentiality guarantee, data owners can encrypt the files and set an access policy so that only qualified users can decrypt the document. With Ciphertext-Policy Attribute-based Encryption (CP-ABE), we can have both fine-grained access control and strong confidentiality. However, this access control is only available for data owners, which turn out to be insufficient. If the cloud provider cannot authenticate users before downloading, like in many existing CP-ABE cloud storage systems, the cloud has to allow everyone to download to ensure availability. This makes the

ISSN- 2394-5125 VOL 10, ISSUE 03, 2023

storage system vulnerable to the resource-exhaustion attacks. If we resolve this problem by having data owners authenticate the downloader's before allowing them to download, we lose the flexibility of access control from CP-ABE. Here lists the two problems should be addressed in our work.

- **Problem I: resource-exhaustion attack:** If the cloud cannot do cloud-side access control, it has to allow anyone, including malicious attackers, to freely download, although only some users can decrypt. The server is vulnerable to resource-exhaustion attacks. When malicious users launch the DoS/DDoS attacks to the cloud storage, the resource consumption will increase. Payers (in pay-as-you-go model) have to pay for the increased consumption contributed by those attacks, which is a considerable and unreasonable financial burden. The attack has been introduced as Economic Denial of Sustainability (EDoS), which means payers are financially attacked eventually. In addition, even files are encrypted, unauthorized downloads can reduce security by bringing convenience to to offline analysis and leaking information like file length or update frequency.
- **Problem II: resource consumption accountability:** In the pay-as-you-go model, users pay money to the cloud provider for storage services. The fee is decided by resource usage. However, CP-ABE based schemes for cloud storage access control does not make online confirmations to the data owner before download. It is needed for the cloud service provider to prove to the payers about the actual resource usage. Otherwise, the cloud provider can charge more without being discovered.

1.2 Summary of Challenges and Approaches

Challenge I: modelling the cloud provider: Many existing CP-ABE based schemes, model the cloud providers (like Google, Amazon, Microsoft Azure) as semi-honest adversaries or passive attackers. However, such a definition is restricted, and it excludes some possible attacks in the real world, such as exaggerated resource usage. To model such attacks, we consider a less restricted security model, covert adversary, for the cloud provider.

In practice, the cloud services are usually provided by some big IT enterprises like Google, Amazon, and Microsoft. They need to maintain good reputation and promise secure cloud storage services to their customers. If any attempt the cloud provider deviates from the protocol is supposed to be caught with a possibility (e.g., $p = 0.001$), the cloud provider dares not to cheat. Because being caught will not only violate the service contracts, but also lead to media exposure and destroys the reputation. Aware of the aftermath, the cloud provider has to refrain from attacking, as the cheating can be detected. This model, covert security, has been used in many secure systems.

Note that the covert security model is different with the semi-honest model. The semi-honest model, which is widely used in proxies and cloud providers, is a model that resides between "malicious" and "trusted". It models a party that observes all data, but it never executes the wrong program. Such a party will not cheat by definition, even if no other parties can detect its cheating. The covert model, which resides between "malicious" and "semi-honest", models this party differently. It will not execute the wrong program only if there is a mechanism to detect its cheating. If no detection exists in the system, the party may even compromise the data, for example. Therefore, it is more practical for public cloud storage.

Approach: model cloud providers as covert adversaries, and design protocols resilient to a covert adversary.

Challenge II: compatible with existing systems

There are many constructions and variants for CP-ABE. We don't design a new variant of CP-ABE to resolve the first challenge, as it is hard to achieve all the functionalities in these systems and also, it's not necessary. Besides the functionalities, some variants provide additional security and privacy guarantee. For example, the literatures hide the access policy. If the cloud-side access control makes

ISSN- 2394-5125 VOL 10, ISSUE 03, 2023

the cloud provider knowing the access policy, it is not considered secure and compatible. It requires the cloud-side access control to be zero-knowledge for arbitrary CP-ABE schemes.

Approach: use CP-ABE in a syntactical and black-box way and ensure the construction not leaking policy and attributes. The system only learns whether the user is legitimate or not, and nothing else.

Challenge III: minimal performance overhead

To protect the cloud storage effectively against the resource-exhaustion attack, the cloud-side access control needs to be efficient and lightweight, otherwise if the cloud server spends, for example 20ms, executing the cloud-side access control, it will become a computational resource exhaustion attack, which can be used by malicious attackers for DDoS and EDoS. The performance overhead being small also benefits the data users who download the files from the cloud storage, making the computation not approachable to resource-limited devices.

Approach: design an efficient access control for the cloud provider which should not add too much overhead.

1.3 Our work and Contribution

In this work, we combine the cloud-side access control and the existing data owner-side CP-ABE based access control, to resolve the aforementioned security problems in privacy-preserving cloud storage. Our method can prevent the EDoS attacks by providing the cloud server with the ability to check whether the user is authorized in CP-ABE based scheme, without leaking other information.

For our cloud-side access control, we use CP-ABE encryption/decryption game as challenge-response. While upload an encrypted file, the data owner firstly generates some random challenge plaintexts and the corresponding ciphertext. The ciphertext are related to the same access policy with the specific file. For an incoming data user, the cloud server asks him/her to decrypt randomly selected challenge ciphertext. If the user shows a correct result, which means he/she is authorized in CP-ABE, the cloud-side access control allows the file download.

To make our solution secure and efficient in real world applications, we provide two protocols of cloud-side and data owner-side combined access control. The main contribution of this work can be summarized as follows.

1) We propose a general solution to secure encrypted cloud storage to prevent the EDoS attacks, as well as have fine-grained access control and resource consumption accountability. To the best of our knowledge, this is the first work to claim that insufficient cloud-side access control in encrypted cloud storage will lead to EDoS attacks and provides a practical solution. The solution can be compatible with many CP-ABE schemes.

2) For different data owner online patterns and performance concern, we provide two protocols for authentication and resource consumption accounting. We also introduce the bloom filter and the probabilistic check to improve the efficiency but still guarantee the security.

3) Compared with many state-of-arts constructions of encrypted cloud storage that assume the existence of a semi-honest cloud provider, we use a more practical threat model where we assume the cloud provider to be a covert adversary, which provides higher security guarantee.

2. LITERATURE SURVEY

To conduct a fine-grained data owner-side access control in public cloud storage, which is semi-honest, Attribute-based Encryption (ABE) [1-3] is introduced [4]. Among ABE schemes, CP-ABE [1], [5] is practical in public cloud storage, in which the ciphertext is encrypted under an access policy and only users whose attributes satisfy the access policy can decrypt the ciphertext. Subsequently, many variants and relevant protocols [6-9] have been proposed to make CP-ABE more suitable for real scenarios with rich functionalities and security properties in public cloud storage.

The cryptography-driven access control does not protect the cloud provider against many other attacks. Since the cloud provider does not conduct the access control, it cannot stop those unauthorized users. One attack that is originated from this limitation is Distributed Denial of Services

ISSN- 2394-5125 VOL 10, ISSUE 03, 2023

(DDoS). The power of DDoS attacks has been showed to incur significant resource consumption in CPU, memory, I/O, and network [10]. The attacks can exist in public clouds [11-14]. In [12], the limitation of cloud-side static resource allocation model is analyzed, including the risk of Economic Denial of Sustainability (EDoS) attacks, which is the case of DDoS attacks in the cloud setting in [14], or the Fraudulent Resource Consumption (FRC) attack in [11]. These attacks are intended to break the budget of public cloud customers.

Some existing works try to mitigate EDoS attacks [15, 16]. In [15], the authors proposed a mitigation technique by verifying whether a request comes from a cloud user or is generated by bots. In [16], the authors proposed an attribute-based way to identify malicious clients. They treat the underlying application in a black box and do not fully immunize the attack in the algorithmic and protocol level. Some existing works discuss the necessary of accounting resource consumption in the public cloud arouses some concerns. In the literature [17], the authors discussed key issues and challenges about how to achieve accountability in cloud computing.

In the literature [18], the authors surveyed existing accounting and accountability in content distribution architectures. In the literatures [19] and [20], the authors respectively proposed a systematic approach for verifiable resource accounting in cloud computing. However, the accounting approach involves changes to the system model, and requires the anonymous verification of users, which is not supported in previous systems. Compared with relevant schemes, our approach works on the protocol level to provide the resource verifiability that relies on authorized users who satisfy the CP-ABE policy and achieves the covert security which is more practical and secure.

3. PROPOSED SCHEME

3.1 Overview of our scheme

To achieve the security requirements, the scheme consists of two components: 1) A cloud-side access control to block users whose attribute set $\mathcal{A}_i$ does not satisfy the access policy A; 2) A proof-collecting subsystem where the cloud provider can collect the proofs of resource consumption from users, and present to the data owners later.

In real-world scenarios, it is reasonable to specify an expected maximal download time, and data owners can remain offline unless it wants to increase this value. This leads to our first protocol: Partially Outsourced Protocol (POP) (V-B). In some other cases where the data owner cannot set expectations of download times or would be offline for a long time, the data owner can delegate to the cloud. This leads to our second protocol: Fully Outsourced Protocol (FOP) (V-C).

3.2 Partially Outsourced Protocol (POP)

In this protocol, the data owner encrypts an ephemeral key in CP-ABE, which is then used for message encryption/decryption and cloud-side access control. The data owner provides the cloud provider with N challenge cipher texts $\{enchal_i\}_{i\in[N]}$ and the hashed challenges$\{hash_i\}_{i\in[N]}$. The user proves the legitimacy to the cloud provider by showing the decryption result $chal_j$ of the randomly selected unused challenge ciphertext $enchal_j$ is a preimage of$hash_j$. If the user response is valid, the cloud provider stores the user response for further resource consumption accounting.

Furthermore, to boost the efficiency and together reduce the storage space, we introduce the bloom filter for data owners to store their challenge plaintexts. This bloom filter can be stored locally or remotely on the cloud server. As the process of challenge update should be implemented on demand or periodically by the data owner, which cannot be outsourced to the cloud, we call the scheme as Partially Outsourced Protocol (POP).

The procedure of POP is described in detail as follows:

1) Encrypt and Upload (POP-EU): This operation is implemented by an individual data owner independently

2) Cloud-side Access Control: POP-CR.

ISSN- 2394-5125 VOL 10, ISSUE 03, 2023

3) Challenge update (POP-SU): If the specified upper bound of download times (N) has not yet reached, there is no need to update. But if the data owner wants to provide additional challenges, either on-demand or periodically, both only needs to be online for a short period, it is also supported. The update process is the same as that in the phase of POP-EU-2 under the same key k. We assume the data owner keeps a record of session keys either in local storage or outsourced to cloud in an encrypted form. As the plaintext space for challenges is sufficiently large, we assume no duplicated challenge plaintexts are generated. The bloom filter (and its encryption form) introduced in POP-EU-3 will be reconstructed.

4) Resource Accounting (POP-RA): data owners and the cloud interactively implement this operation.

3.3 Fully Outsourced Protocol (FOP)

If we cannot expect the file download times, we can outsource the challenge update to the cloud. In this section, we give a protocol based on signature algorithm, which has both the outsourced challenges generation/update and resource accounting without an external PKI, therefore we name it as Fully Outsourced Protocol (FOP).

Compared with POP, we have two main differences: 1) Instead of having the data owners generate the challenges$\{enchal_i\}_{i\in[N]}$, the challenges are generated by the cloud; 2) The data owners generate a pair of signature keys (vk,sk) for every file, with which legitimate users sign a confirmation to prove the resource consumption. The main procedure of FOP is described as follows:

1) Encrypt and Upload (FOP-EU)

2) Outsourced Challenge Generation (FOP-CG): In FOP, the cloud provider generates the challenges, which is different from POP. The generation can be done in advance or on demand.

3) Challenge-Response (FOP-CR). Data owners and the cloud run this operation

4) Resource Accounting (FOP-RA). This operation is interactively implemented by the data owner and the cloud.

3.4 System modules

Cloud Computing allow users to store or access data from anywhere and anytime with cheap cost. All data storage at cloud side will be at security risk due to unavailable control of data owner on store data. To provide security to data many data security algorithms are introduce and the most famous one is CP-ABE (Cipher Policy Attribute Based Encryption). In this algorithm data owner can encrypt data by specifying attributes of those users who can access data and the CP-ABE will generate encryption public and private keys by using those attributes and then encrypt and upload data to cloud. Any user with access control can request file from the cloud and then download that file and if user has permission in his attributes, then file will be decrypted otherwise file will not be decrypted. With this algorithm access control and data security can be achieve but the drawback of CP-ABE is first it allows user to download file whether it has permission or not and after download he can decrypt the file if he has permission. Due to file downloading prior decryption can raise Economical Denial of Sustainability (EDoS) attack. In this attack malicious users will download files (attackers know they cannot decrypt file but still to raise problems they will download files to make cloud busy and put charges on customers) and consume cloud resources and this consumption charges will be applied on customers.

To avoid author has introduce concept called Combining Data Owner & Cloud Side Access Control. In this technique while uploading file user will generate secret data and encrypt that secret data with bloom filter algorithm and then encrypt file data with CP-ABE and then upload encrypted file with secret data and bloom filter data to cloud for storage. If any user wants to download file, then cloud will ask secret data from user and then encrypt that data with bloom filter and check existing data

ISSN- 2394-5125 VOL 10, ISSUE 03, 2023

owner bloom filter with user bloom filter and if match found then only cloud send download file to user.

By applying secret data bloom filter match author has prevented EDoS attack, author introduce two technique two avoid EDoS attack.

1) POP (Partially Outsource Protocol): using this technique cloud allow data owner to generate secret data for user verification before file download.
2) FOP (Fully Outsource Protocol): using this technique instead of user cloud will generate secret data with bloom filter for user verification before download file.

This paper consists of 3 users.

3.4.1 Data Owner: data owner will upload file and then using CP-ABE define access control and then encrypt data and then outsource encrypted data with secret key data for user verification. Sometime cloud may cheat customers by saying customer has consume this many resources and the author is saying big companies may not do that but still to prevent cloud from fraud usage cost author has provided customer an option to verify resource consumption. By using this option data owner can request cloud to provide details about his data usage or download.

3.4.2 Data User: this is the user of data which request cloud for file download and before download cloud will ask user for verification by entering secret data obtained from data owner. All data owners share their secret data with their data users.

3.4.3 Cloud Provider: This is a cloud server which store user data and performs user verification and provides resource consumption details to data owners.

For encryption and decryption, we are using CPABE algorithm. Below steps are involved to key generation, encrypt and decrypt data using CPABE.

Attributes = "user1, user2, user3" //list of users assign to attribute variables who have access permission

Cpabe att = new Cpabe(); //creating object of CPABE class

String public_key = file.getPath()+"/public.txt"; //file used to store public key

String master_key = file.getPath()+"/master.txt"; //files used to store master key

String private_key = file.getPath()+"/private.txt"; //file used to store private key

att.setup(public_key,master_key); //here cpabe oject initialize with public and master key file

att.keygen(public_key,private_key,master_key,attributes); //here cpabe object generate key based on given attributes. Attributes contains names of users who are having permission to access data after generating keys then user call encrypt function to encrypt data by using keys and attributes data. Below code used to achieve encryption

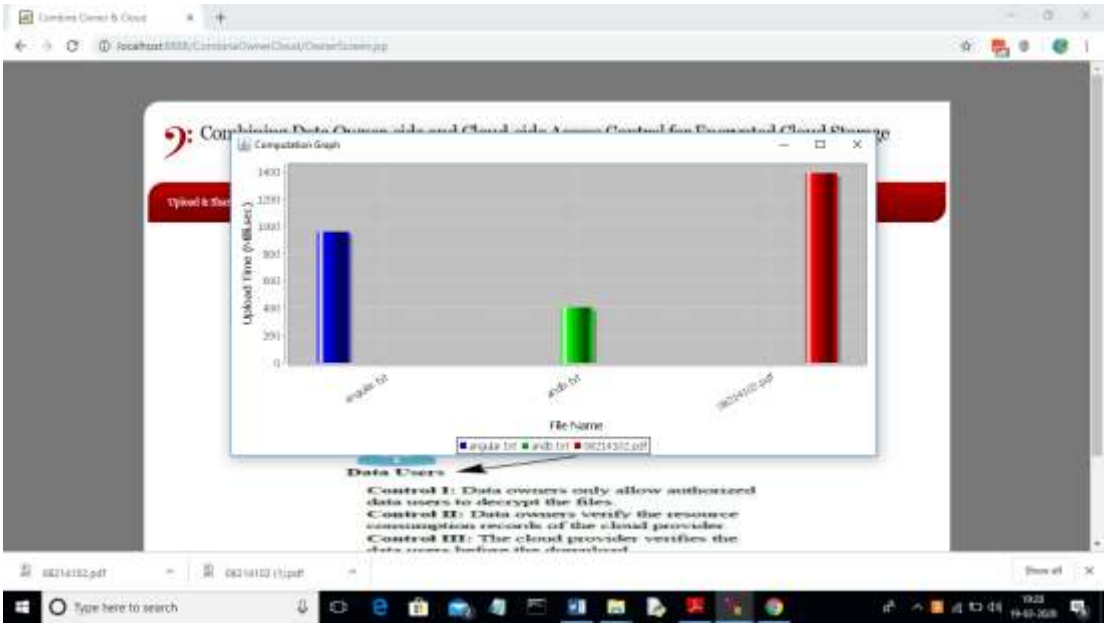
Cpabe att = new Cpabe(); //creating object of CPABE class

String public_key = input; //before encrypting giving name and location public key file att.enc(public_key,policy,encfile,encdata); //now calling encryption function by using public key and policy file (this also contains peoples id who have permission to access data), encfile is the plain file which will get encrypted and then store to encdata file while decryption will used below code Cpabe test = new Cpabe();//creating object of CPABE class test.dec(public_key,private_key,enc,dec.getPath()); //calling dec function to decrypt data while calling this function we will pass private key of person who is accessing data and enc is the encrypted data and dec.getPath() is the file location to which encrypted data will be decrypted and save to dec.getPath()

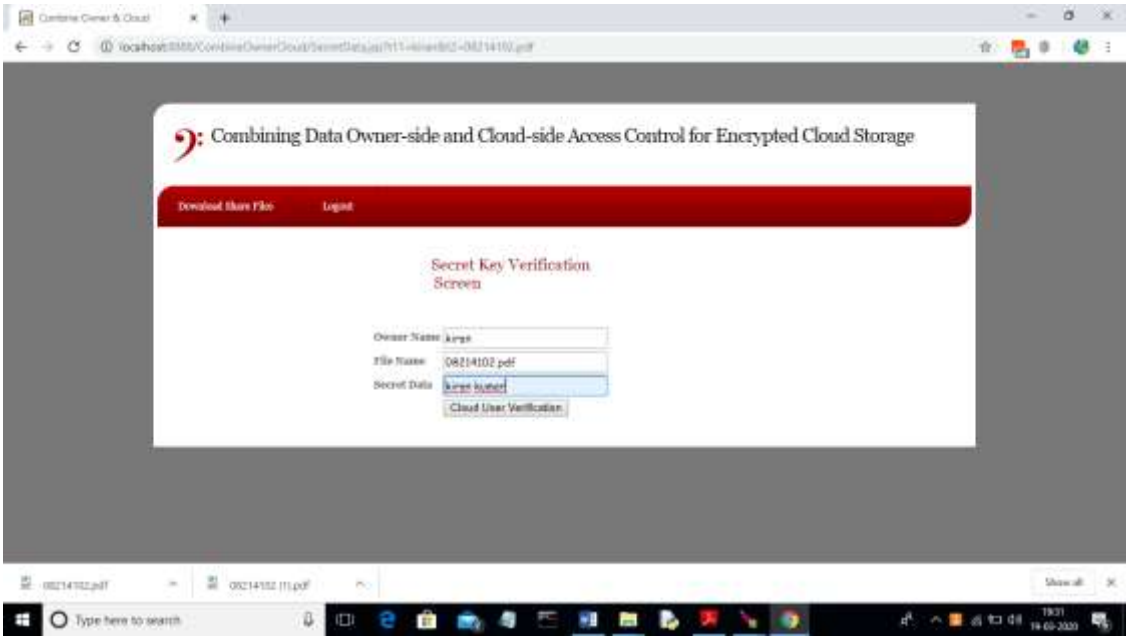
So using above techniques we are encrypting and decrypting data and secret data with bloom filter can be applied by using algorithm called AEAD and secret data also called as CHALLENGE.

4. RESULTS

ISSN- 2394-5125 VOL 10, ISSUE 03, 2023

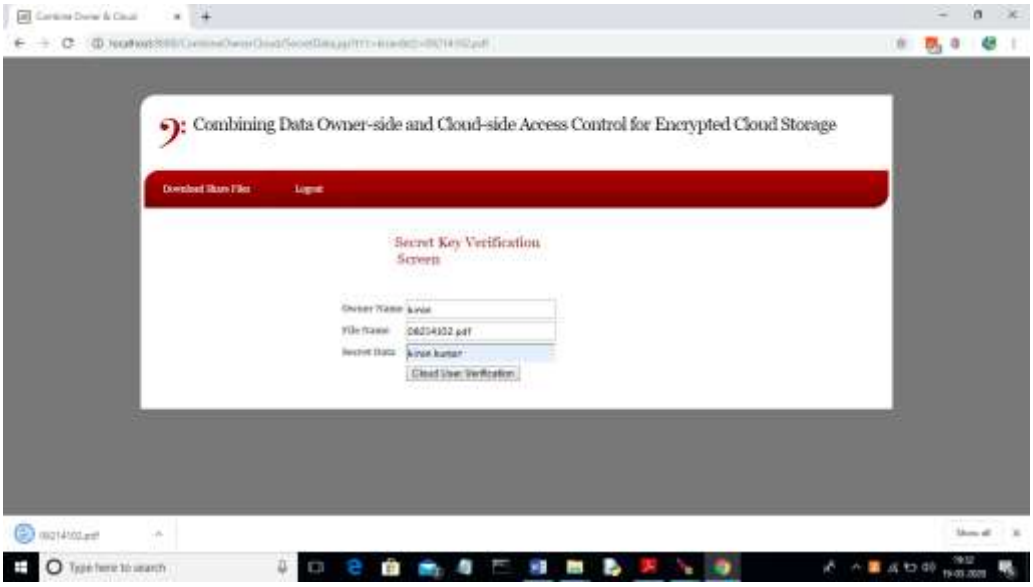


In above graph x-axis represents file name and y-axis represents execution time to encrypt and upload that file to cloud.

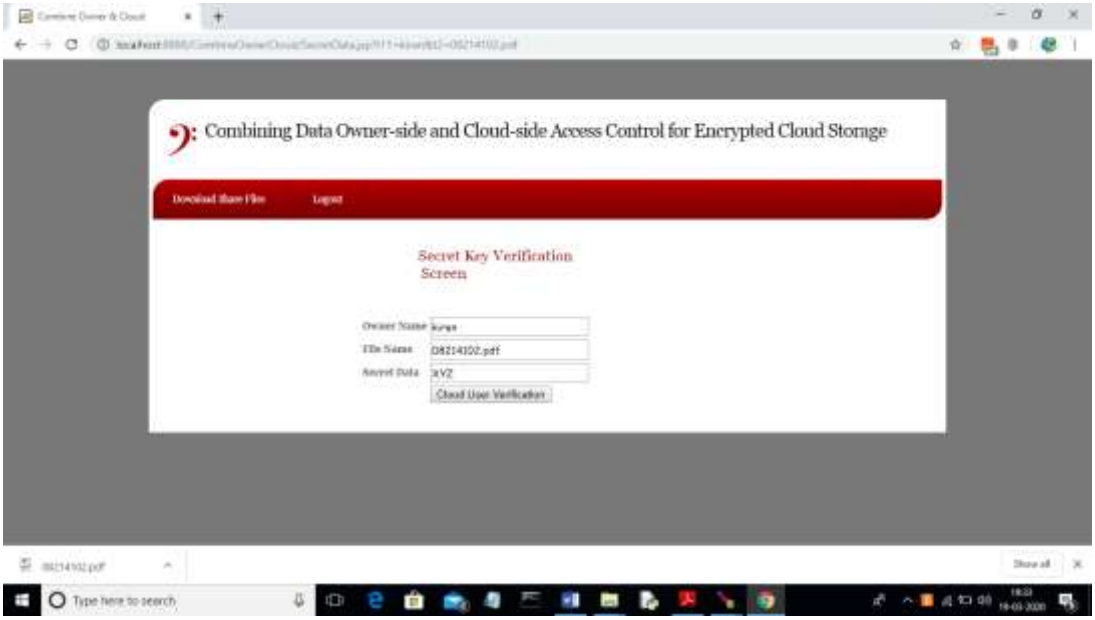


In above screen cloud is asking data user for secret data challenge and if user give correct data owner secret data then only file will be downloaded otherwise not

ISSN- 2394-5125 VOL 10, ISSUE 03, 2023

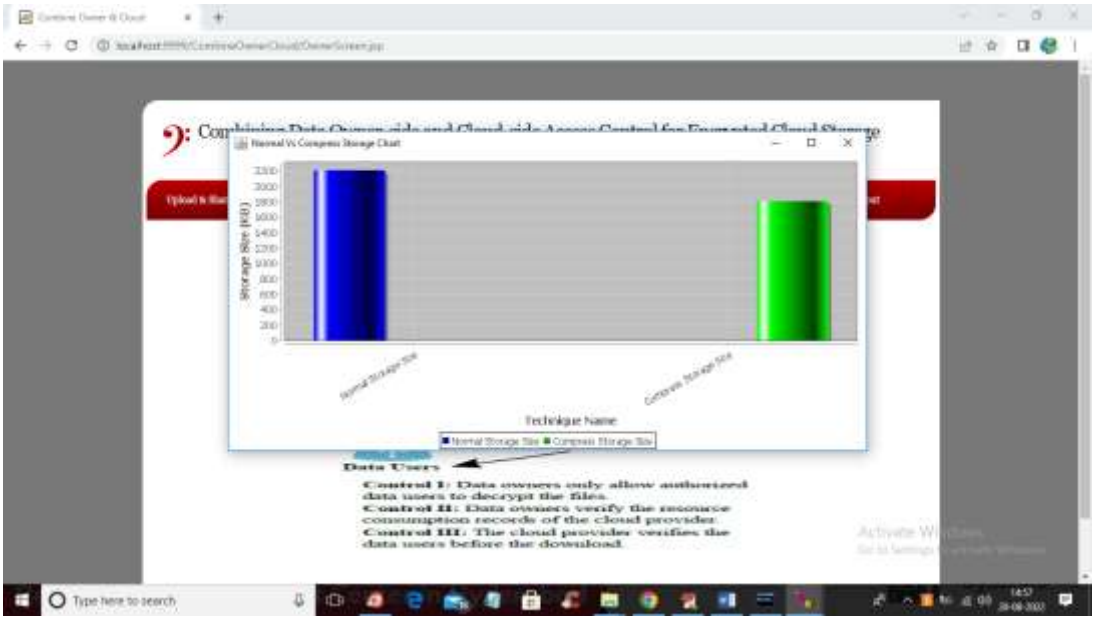


In above screen I am giving correct secret data and we can see file downloaded in browser status bar and in below screen I will give wrong secret data



After giving wrong secret challenge will get below screen.

In addition, this project added compression technique which will compress all files before storing cloud. As we know all Cloud Service Providers will charge customers based on usage so we need to use resources efficiently and effectively so cloud charges can be reduced. If we store huge files, then cloud will take more storage space and this space can be reduced by applying compression technique which will compress data by replacing redundant data with its position INDEX. So, by removing redundant data we can reduce file size and cloud storage space.



In the above comparison graph x-axis represents Normal and Compress Storage and y-axis represents Memory Space taken by both techniques and from above graph we can see say by storing compress files we can save more space at cloud.

5. CONCLUSION

This paper proposed a solution to secure encrypted cloud storages from EDoS attacks. In addition, it also provided resource consumption accountability. Further, it utilized CP-ABE schemes in a black-box manner and complies with arbitrary access policy of CP-ABE. Finally, this paper presented two protocols for different settings, followed by performance and security analysis.

REFERENCES

[1] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in 2007 IEEE Symposium on Security and Privacy (SP'07). IEEE, 2007, pp. 321–334.

[2] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in Advances in Cryptology–EUROCRYPT 2005. Springer, 2005, pp. 457–473.

[3] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proceedings of the 13th ACM conference on Computer and communications security (CCS2006). ACM, 2006, pp. 89–98.

[4] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in The 29th IEEE International Conference on Computer Communications (IEEE INFOCOM 2010). IEEE, 2010, pp. 1–9.

[5] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in Public Key Cryptography– PKC 2011. Springer, 2011, pp. 53–70.

[6] W. Li, K. Xue, Y. Xue, and J. Hong, "TMACS: A robust and verifiable threshold multi-authority access control system in public cloud storage," IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 5, pp. 1484–1496, 2016.

[7] T. V. X. Phuong, G. Yang, and W. Susilo, "Hidden ciphertext policy attribute-based encryption under standard assumptions," IEEE Transactions on Information Forensics and Security, vol. 11, no. 1, pp. 35–45, 2016.

[8] K. Yang, X. Jia, and K. Ren, "DAC-MACS: Effective data access control for multi-authority cloud storage systems," in Proceedings of the 32nd IEEE International Conference on Computer Communications (Infocom2013). IEEE, 2013, pp. 2895–2903.

[9] K. Xue, Y. Xue, J. Hong, W. Li, H. Yue, D. S. Wei, and P. Hong, "RAAC: Robust and auditable access control with multiple attribute authorities for public cloud storage," IEEE Transactions on Information Forensics and Security, vol. 12, no. 4, pp. 953–967, 2017.

[10] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," ACM Computing Surveys, vol. 39, no. 1, p. 3, 2007.

[11] J. Idziorek and M. Tannian, "Exploiting cloud utility models for profit and ruin," in Proceedings of 2011 IEEE International Conference on Cloud Computing (CLOUD2011). IEEE, 2011, pp. 33–40.

[12] S. Yu, Y. Tian, S. Guo, and D. O. Wu, "Can we beat DDoS attacks in clouds?" IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 9, pp. 2245–2254, 2014.

[13] Q. Chen, W. Lin, W. Dou, and S. Yu, "CBF: a packet filtering method for DDoS attack defense in cloud environment," in Proceedings of IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC2011). IEEE, 2011, pp. 427–434.

[14] C. Hoff, "Cloud computing security: From DDoS (distributed denial of service) to EDoS (economic denial of sustainability)," http://www. rationalsurvivability.com/blog/?p=66.

[15] M. H. Sqalli, F. Al-Haidari, and K. Salah, "EDoS-Shield - a twosteps mitigation technique against edos attacks in cloud computing," in Proceedings of 4th IEEE International Conference on Utility and Cloud Computing (UCC2011). IEEE, 2011, pp. 49–56.

[16] J. Idziorek, M. Tannian, and D. Jacobson, "Attribution of fraudulent resource consumption in the cloud," in Proceedings of the 5th IEEE International Conference on Cloud Computing (CLOUD2012). IEEE, 2012, pp. 99–106.

[17] R. K. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang, and B. S. Lee, "TrustCloud: A framework for accountability and trust in cloud computing," in 2011 IEEE World Congress on Services (SERVICES 2011). IEEE, 2011, pp. 584–588.

[18] D. O. Coile ´ ain and D. O'mahony, "Accounting and accountability in ´ content distribution architectures: A survey," ACM Computing Surveys (CSUR), vol. 47, no. 4, p. 59, 2015.

[19] V. Sekar and P. Maniatis, "Verifiable resource accounting for cloud computing services," in Proceedings of the 3rd ACM workshop on Cloud computing security workshop. ACM, 2011, pp. 21–26.

[20] C. Chen, P. Maniatis, A. Perrig, A. Vasudevan, and V. Sekar, "Towards verifiable resource accounting for outsourced computation," in ACM SIGPLAN Notices, vol. 48, no. 7. ACM, 2013, pp. 167–178.