AN IMPROVED SECURE SEARCHABLE ENCRYPTION FRAMEWORK FOR CLOUD STORAGE SERVICES

Rakeshkumar.P[1], Mr.A.G.Ignatius.M.E/A.P.[2]
[1,2] Dept of Information and Technology
[1,2] Sri Muthukumaran Institute of Technology

Abstract- *In this paper, we propose a novel solution called key-policy attribute based encryption with time-specified attributes scheme, which is based on our observation that, in practical cloud application scenarios, each data item can be associated with a set of attributes and every attribute is associated with a specification of time interval e.g., [09:00,17:00], noting that the encrypted data item can only be decrypted between 09:00 to 17:00 on a specified date and it will not be recoverable before 09:00 and after 17:00 that day. The data owner encrypts his/her data to share with users in the system, in which every user's key is associated a time instant, e.g., 14:30. The access tree of each user can be defined as a unique logical expression over these DATI attributes to reflect the data item authorized to the user. In order to decrypt the cipher text successfully, the valid attributes should satisfy the access tree where the time instant of each leaf in the users key should belong to the DATI (e.g., 14:30 2 [09:00,17:00]) in the corresponding attribute in the cipher text . As the logical expression of the access tree can represent any desired data set with any time interval, it can achieve fine-grained access control. If the time instant is not in the specified time interval, the cipher text cannot be decrypted, i.e., this cipher text will be self-destructed and no one can decrypt it because of the expiration of the secure key. Therefore, secure data self-destruction with fine-grained*

Keywords- Cloud Storage, Self Destructing

I. INTRODUCTION

Accessible encryption has gotten a huge consideration from the exploration network with different developments being proposed, each accomplishing asymptotically ideal multifaceted nature for explicit measurements (e.g., search, update). In spite of their class, the ongoing assaults and sending endeavors have demonstrated that the ideal asymptotic multifaceted nature may not generally suggest down to earth execution, particularly if the application requests a high security. We present a novel Dynamic Searchable Symmetric Encryption (DSSE) system called Incidence Matrix (IM)-DSSE, which accomplishes a significant level of security, productive inquiry/update, and low customer stockpiling with genuine organizations on genuine cloud settings. We bridle an occurrence framework alongside two hash tables to make an encoded file, on which both pursuit and update activities can be performed successfully with negligible data spillage. This straightforward arrangement of information structures shockingly offers an elevated level of DSSE security while accomplishing down to earth execution. In particular, IM-DSSE accomplishes forward-protection, in reverse security and size-mindlessness at the same time. We likewise make a few DSSE variations, each offering diverse exchange offs that are appropriate for various cloud applications and frameworks. We completely executed our structure and assessed its exhibition on a genuine cloud framework (Amazon EC2). We have discharged IM-DSSE as an open-source library for wide advancement and adjustment.

II. OBJECTIVE

This project represents IM-DSSE, a new DSSE framework which offers very high privacy, efficient updates, low search latency simultaneously.

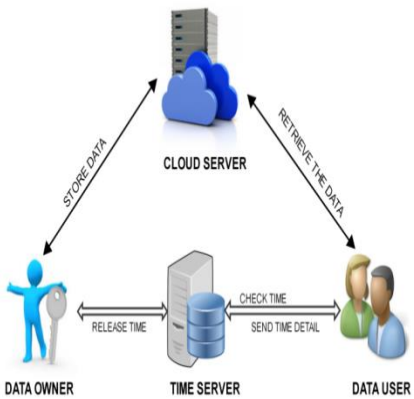
III. LITERATURE SURVEY

Searchable encryption has received a significant attention from the research community with various constructions being proposed, each achieving asymptotically optimal complexity for specific metrics (e.g., search, update). Despite their elegance, the recent attacks and deployment efforts have shown that the optimal asymptotic complexity might not always imply practical performance, especially if the application demands a high privacy. In this article, we introduce a novel Dynamic Searchable Symmetric Encryption (DSSE) framework called Incidence Matrix (IM)-DSSE, which achieves a high level of privacy, efficient search/update, and low client storage with actual deployments on real cloud settings. We harness an incidence matrix along

with two hash tables to create an encrypted index, on which both search and update operations can be performed effectively with minimal information leakage. This simple set of data structures surprisingly offers a high level of DSSE security while achieving practical performance. Specifically, IM-DSSE achieves forward-privacy, backward-privacy and size-obliviousness simultaneously. We also create several DSSE variants, each offering different trade-offs that are suitable for different cloud applications and infrastructures. We fully implemented our framework and evaluated its performance on a real cloud system (Amazon EC2). We have released IM-DSSE as an open-source library for wide development and adaptation.

Recently, several practical attacks raised serious concerns over the security of searchable encryption. The attacks have brought emphasis on forward privacy, which is the key concept behind solutions to the adaptive leakage-exploiting attacks, and will very likely to become a must-have property of all new searchable encryption schemes. For a long time, forward privacy implies inefficiency and thus most existing searchable encryption schemes do not support it. Very recently, Bost (CCS 2016) showed that forward privacy can be obtained without inducing a large communication overhead. However, Bost's scheme is constructed with a relatively inefficient public key cryptographic primitive, and has poor I/O performance. Both of the deficiencies significantly hinder the practical efficiency of the scheme, and prevent it from scaling to large data settings. To address the problems, we first present FAST, which achieves forward privacy and the same communication efficiency as Bost's scheme, but uses only symmetric cryptographic primitives. We then present FASTIO, which retains all good properties of FAST, and further improves I/O efficiency. We implemented the two schemes and compared their performance with Bost's scheme. The experiment results show that both our schemes are highly efficient.

BLOCK DIAGRAM:



IV.MODULES

File uploading and activation:The data owner firstly needs to generate signatures for data blocks before uploading them to the cloud. These signatures are used to prove the cloud truly possesses these data blocks in the phase of integrity auditing. And then the data owner uploads these data blocks along with their corresponding signatures to the cloud. The data stored in the cloud is often shared across multiple users in many cloud storage applications. The data owner activate the file to check whether the uploaded file is appropriate or not then the Proxy also activate the file to check the file is Good.

Data integrity auditing: Data integrity auditing scheme that realizes data sharing with sensitive information hiding. However, the data stored in the cloud might be corrupted or lost. Data integrity auditing on the condition that the sensitive information of shared data is protected.

Sensitive information sharing: Sensitive information hiding to ensure that the personal sensitive information of the file is not exposed to the hacker and all of the sensitive information of the file is not exposed to the cloud and the shared users. This method not only realizes the remote data integrity auditing, but also supports the data sharing on the condition that sensitive information is protected in cloud storage.

Generating key signature: A potential method of solving this problem is to encrypt the whole shared file before sending it to the cloud, and then generate the signatures used to verify the integrity of this encrypted file, finally upload this encrypted file and its corresponding signatures to the cloud. This method can realize the sensitive information hiding since only the data owner can decrypt this file. However, it will make the whole shared file unable to be used by others.

File security: If a file has been partially overwritten or otherwise compromised, the chances of any usable recovery are low, even with the best recovery software in the existing system. In our proposed work, we can easily recover the file while deleted files are inaccessible and are in danger of being overwritten, they can often be recovered.

V. ADVANTAGES

- This project introduce a Dynamic Searchable Symmetric Encryption (DSSE) framework called Incidence Matrix (IM)-DSSE, which achieves a high level of privacy, efficient search/update, and low client storage with actual deployments on real cloud settings.

- This project harness an incidence matrix along with two hash tables to create an encrypted index, on which both search and update operations can be performed effectively with minimal information leakage.
- This also create several DSSE variants, each offering different trade-offs that are suitable for different cloud applications and infrastructures

VI. DISADVANTAGES

- Preliminary SSE schemes only provide search only functionality on static data (i.e., no dynamism), which strictly limits their applicability due to the lack of update capacity.
- IM-DSSE allows to directly updating keywords of an existing file without invoking the file delete-then-add operation sequence. The update in IM-DSSE also leaks minimal information

CONCLUSION

We displayed IM-DSSE, another DSSE system which offers extremely high protection, proficient updates, and low search idleness at the same time. Our developments depend on a basic yet effective rate framework information structure in blend with two hash tables that permit effective and secure pursuit and update activities. Our structure offers different DSSE developments, which are explicitly structured to address the issues of cloud foundation and individual use in various applications and situations. The entirety of our plots in IM-DSSE system is demonstrated to be secure and accomplish the most noteworthy protection among their partners. We directed a nitty gritty trial investigation to assess the execution of our plans on genuine Amazon EC2 cloud frameworks. Our outcomes demonstrated the high common sense of our system, in any event, when conveyed on cell phones with enormous datasets. We have discharged the undeniable usage of our system for open use and investigation.

REFERENCES

[1] K. Ren, C. Wang, and Q. Wang, “Security challenges for the public cloud,” IEEE Internet Computing, vol. 16, no. 1, pp. 69–73, Jan 2012.
[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable data possession at untrusted stores,” in Proceedings of the 14th ACM Conference on Computer and Communications Security, ser. CCS ’07, 2007, pp. 598–609.
[3] A. Juels and B. S. Kaliski, “Pors: Proofs of retrievability for large files,” in Proceedings of the 14th ACM Conference on Computer and Communications Security, ser. CCS ’07, 2007, pp. 584–597.
[4] H. Shacham and B. Waters, “Compact proofs of retrievability,” J. Cryptology, vol. 26, no. 3, pp. 442–483, Jul. 2013.