A Secure Searchable Encryption Framework For Privacy-Critical Cloud Storage Services

N.SRINIVASA RAO [1], DANGETI NAGA MOUNICA.

[1] Assistant Professor, DEPT OF MCA, SKBR PG COLLEGE, AMALAPURAM, Andhra Pradesh

Email:- naagaasrinu@gmail.com

[2]PG Student of MCA, SKBR PG COLLEGE, AMALAPURAM, Andhra Pradesh

Email:- dangetinagamounika@gmail.com

ABSTRACT With numerous constructions being proposed, searchable encryption has gained considerable interest from the scientific community, each reaching asymptotically optimal difficulty for particular metrics (e.g., search, update). The recent attacks and implementation attempts have demonstrated, considering their sophistication, that the optimum asymptotic complexity could not often equal realistic efficiency, particularly if the application involves a high degree of privacy. In this paper, we present Occurrence Matrix (IM)-DSSE, a modern Dynamic Searchable Symmetric Encryption (DSSE) architecture that achieves a high degree of safety, fast search/update, and low customer storage with actual implementations on real cloud settings. To construct an encrypted index, we use an incidence matrix along with two hash tables, on which both search and upgrade operations can be conducted efficiently with minimal leakage of knowledge. This basic collection of data structures, while achieving realistic efficiency, surprisingly provides a high degree of DSSE protection. Specifically, IM-DSSE concurrently achieves forward-privacy, backwardprivacy and size-obliviousness. We also develop many DSSE versions, each of which provides numerous trade-offs appropriate for various cloud applications and infrastructures. We have completely applied our architecture and assessed its performance on a real cloud environment (Amazon EC2). As an open-source library for comprehensive creation and adaptation, we have published IM-DSSE.

I. INTRODUCTION

For society and the IT sector, the growth of cloud infrastructure and computer facilities brings tremendous benefits. Data Storage-as-a-Service (SaaS) is one of the most relevant cloud technologies, and can dramatically minimise data processing costs by continuous service, experience and support for resource-limited users such as individuals or small/medium enterprises. Given its advantages, SaaS often introduces to the consumer essential protection and privacy issues. That is, once a client outsources its own data to the cloud, a malicious party may manipulate confidential information (e.g., email) (e.g., malware). While basic encryption schemes may offer secrecy, such as the Advanced Encryption Standard (AES), they also prohibit the client from querying encrypted data from the cloud. The advantages and efficiency of cloud systems can be substantially degraded by this privacy versus data usage problem. It is therefore important to build privacy-enhancing solutions that, though maintaining the practicality of the underlying cloud service, will resolve this problem. Searchable Symmetric Encryption (SSE)[1] helps a device to encrypt data in such a manner that keyword searches can be done on it later. This encrypted queries are performed using "search tokens"[2] over an encrypted index reflecting the relationship between the encrypted files and the search token (keywords). A popular SSE application is to allow cloud keyword search to protect privacy (e.g., Amazon S3), where a data owner may outsource a list of

protected files and conduct keyword searches without exposing the contents of the file and query[3]. Preliminary SSE schemes (e.g.,[1],[4]) only have search-only static data capabilities (i.e. no dynamism), which, due to the lack of updating technology, specifically restricts their applicability. Several Dynamic Searchable Symmetric Encryption (DSSE) systems (e.g.[3],[5]) were subsequently suggested to enable the consumer to add and uninstall files after setting up the device. To the best of our understanding, in terms of all the above parameters, there is no particular DSSE scheme that outperforms all the other alternatives: safety (e.g., information leakage), consistency (e.g., scan, upgrade delay), reliability and functionality of storage. We first give a summary of DSSE research in the following, and then outline our research goals and contributions to resolve some of the state-of-the-art limitations. SSE was presented by Song et al.[4] for the first time. Curtmola et al.[1] suggested a sublinear SSE framework and implemented the SSE protection principle named adaptive protection against selected keyword attacks (CKA2). Refinements to [1] providing expanded features have been suggested (e.g., [6], [7]). The static existence of such systems, however, restricted their applicability to applications involving dynamic arrays of files. Kamara et al. is among the first to build a DSSE scheme in[3] that could accommodate an encrypted index for dynamic file collections. It leaks major data for changes, though, and it is not parallelizable. A DSSE scheme was suggested by Kamara et al.[8], which leaked fewer data than that of[3] and was parallelizable. A variety of new DSSE schemes (e.g., [2], [5], [9], [10], [11], [12]) have recently been suggested that give different trade-offs between security, functionality and efficiency properties, such as low leakage (e.g., [2]), extended query type scalable searches (e.g., [12], [13], [14], [15]), or high efficiency searches (e.g., [12], [13], [14], [15]). (e.g., [9]). Inspired by the work from[5], a new sublinear DSSE scheme was proposed by Kamara et al. in[12] that supports more complicated queries such as disjunctive and boolean queries

II. EXISTING SYSTEM

Many search specific functions such as conjunctive keyword search, disjunctive keyword search and subset search may be carried out utilising the current multi-keyword search systems. Ballardetal. Ballardetal. The proposal includes two separate conjunctive keyword search schemes which, on the basis of the Shamir secret sharing and bilinear pairings, only return the files with all the keywords checked. Their scheme in the regular model has proved secure. And the disjunctive keyword search scheme, which can return files with the subset of question keywords, was consequently proposed. Meanwhile, all conjunctive keyword search and disjunctive keyword search mechanisms are often presented to help predicate encryption.

DIS-ADVANTAGES

1. The data owner has to rebuild the search index tree, which is time-consuming.
2. Traditional solutions have to suffer high computational costs

III. PROPOSED SYSTEM

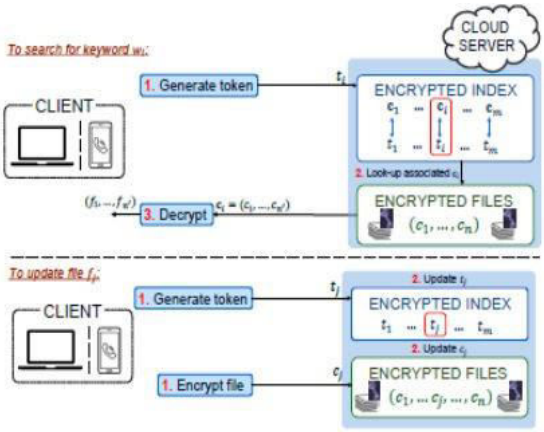
We would suggest a stable, powerful multi-keyword search mechanism to effectively help upgrade operations. The Bloom filter-based index tree can increase the search performance. And our method uses a space vector model to construct an index vector for each file in the outsourcing data collection. The cosine similarity measurement measures the similarity of one file with the search query and uses TF TRIPOIDF weight to maximise search accuracy.

ADVANTAGE

1. Support dynamics properly and reliably.

2. This scheme upgrades the reduced machine costs. 3. Supports complex operations involving record deletions or insertions

IV. ARCHITECTURE



V. IMPLEMENTATION& RESULTS

Data Owner

The data owner uploads its data to the cloud service in this module. The data owner encrypts the data file and only stores it in the cloud for protection purposes. The owner of the data should manage the protected data code. And the data owner may set the right of access to the protected data log.

Towards Cloud Server

The cloud infrastructure company maintains a cloud for data collection. Data owners encrypt and archive their data files in the cloud to connect with data users. Data consumers import encrypted data from the cloud to access the mutual data files and decrypt them. It is liable for all end users' authorisation.

• Hub for Primary Delivery KDC is dedicated to stored authentication parameters and to provide public query services such as generation of a secret key based on the file and submitting it to the corresponding end users. It is in charge of catching the terrorists. Data Consumer/End Usage Data In this module, the user may only access the encrypted key data file if the user has the right to access the file. For the consumer stage, the data owner confers all the rights and the data consumers are only managed by the data owner. Users will either attempt to access data files in their rights of entry, or malicious users may combine together to get critical files outside their privileges. He sends a hidden key request to KDC and KDC will produce the key and submit to the right end-user.

• The attacker (Unauthorized User) Attacker applies malicious data to a cloud server block. The unwanted person is therefore deemed an intruder.

VI. CONCLUSION

In this post, we implemented IM-DSSE, a modern DSSE system that simultaneously provides high privacy, fast alerts, low search latency. Our structures depend on a simple but effective incidence matrix data structure, coupled with two hash tables that render search and update efficient and stable. Our framework provides different DSSE constructions, designed specifically for cloud infrastructure and personal use in various applications and environments. All our IM-DSSE systems have proven safe and have the highest degree of privacy amongst their peers. A detailed experimental analysis was carried out to assess the performance of our schemes on real Amazon EC2 cloud systems. Our results showed that our framework is highly practical even when deployed with large datasets on mobile devices. We have published the full implementation of our public use and analysis framework.

REFERENCES

[1] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: improved definitions and efficient constructions,” in Proc. 13th ACM Conf. Comput. Commun. security, ser. CCS ’06. ACM, 2006, pp. 79–88. [2] E. Stefanov, C.

Papamanthou, and E. Shi, “Practical dynamic searchable encryption with small leakage,” in 21st Annu. Network and Distributed System Security Symp. — NDSS 2014. The Internet Soc., February 23-26, 2014. [3] S. Kamara, C. Papamanthou, and T. Roeder, “Dynamic searchable symmetric encryption,” in Proc. 2012 ACM Conf. Comput. Commun. security. New York, NY, USA: ACM, 2012, pp. 965–976. [4] D. X. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in Proc. 2000 IEEE Symp. Security and Privacy, 2000, pp. 44–55. [5] D. Cash, J. Jaeger, S. Jarecki, C. Jutla, H. Krawcyk, M.-C. Rosu, and M. Steiner, “Dynamic searchable encryption in very-large databases: Data structures and implementation,” in 21th Annu. Network Distributed System Security Symp. — NDSS 2014. The Internet Soc., February 23-26, 2014. [6] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, “Privacy-preserving multi-keyword ranked search over encrypted cloud data,” IEEE Trans. Parallel Distributed Syst., vol. 25, no. 1, pp. 222–233, 2014. [7] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, “Verifiable privacy-preserving multikeyword text search in the cloud supporting similarity-based ranking,” IEEE Trans. Parallel Distributed Syst., vol. 25, no. 11, pp. 3025–3035, 2014. [8] S. Kamara and C. Papamanthou, “Parallel and dynamic searchable symmetric encryption,” in Financial Cryptography and Data Security (FC), ser. Lecture Notes in Comput. Sci. Springer Berlin Heidelberg, 2013, vol. 7859, pp. 258–274. [9] M. Naveed, M. Prabhakaran, and C. A. Gunter, “Dynamic searchable encryption via blind storage,” in 35th IEEE Symp. Security Privacy, May 2014, pp. 48–62. [10] F. Hahn and F. Kerschbaum, “Searchable encryption with secure and efficient updates,” in Proc. 2014 ACM SIGSAC Conf. Comput. and Commun. Security. ACM, 2014, pp. 310–320.