

 (Enriching the Research)	Open Access Research Article
	Volume: 23 Issue: 07
	July, 2023

A SECURE SEARCHABLE ENCRYPTION FRAMEWORK FOR PRIVACY CRITICAL CLOUD STORAGE SERVICES

Mr. K. Gopala Reddy¹, K. Pavani², K. Bindu Sri³, G. Ganesh⁴, K. Teja Prasanna Kumar⁵

¹ Associate Professor, Department of CSE, Ramachandra College of Engineering, Eluru, A.P

^{2,3,4,5} UG Students, Department of CSE, Ramachandra College of Engineering, Eluru, A.P.

ABSTRACT

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. We have released IM-DSSE as an open-source library for wide development and adaptation.

1. INTRODUCTION

Searchable symmetric encryption (SSE) allows one to store data at an untrusted server and later search the data for records (or documents) matching a given keyword while maintaining privacy. Dynamic Searchable Symmetric Encryption (DSSE) enables a client to perform keyword queries and update operations on the encrypted file collections. DSSE scheme that achieves the highest privacy among all compared alternatives with low information leakage, non-interactive and efficient updates, compact client storage, low server storage for large file-keyword pairs with an easy design and implementation. Desirable properties of a practical SSE scheme are as follows:

Dynamism: - It should permit adding new files or deleting existing files from the encrypted file collection securely after the system set-up.

Computational Efficiency and Parallelization: - It should offer fast search/updates Moreover, which are parallelizable across multiple processors.

Communication Efficiency: - Non-interactive update/search operations should be possible to avoid the delays, with a minimum amount of data transmission.

Security: - The information leakage due to search/update operations must be precisely quantified based on formal SSE security notions.

	Open Access Research Article
	Volume: 23 Issue: 07
	July, 2023

Private-key Searchable Encryption: - In the setting of searching on private-key-encrypted data, the user himself encrypts the data. The data and the additional data structures can then be encrypted and stored on the server so that only someone with the private key can access it.

Public-key Searchable Encryption: - In the setting of searching on public-key-encrypted data, users who encrypt the data (and send it to the server) can be different from the owner of the decryption key. DSSE leakage implies a strong property called forward privacy. If we search for a keyword w and later add a new document containing keyword w , the server does not learn that the new document has a keyword we searched for in the past. It also implies backward privacy, namely queries cannot be executed over deleted documents. Apart from the search, access and size patterns, it also leaks (during searches) the document identifiers that were deleted in the past and match the keyword. As such, our scheme achieves forward privacy.

Our DSSE scheme is the first one to support dynamic keywords. As opposed to previous DSSE schemes that require storing information about all the possible keywords that may appear in the documents (i.e., all the dictionary), our scheme stores only information about the keywords that currently appear in the documents.

One of the most important cloud services is data Storage-as-a-Service (SaaS), which can significantly reduce the cost of data security and privacy concerns to the user. That is, once a client out source his/her own data to the cloud, sensitive information (e.g., email) might be exploited by a malicious party (e.g., malware). Although standard encryption schemes such as Advanced Encryption Standard (AES) can provide confidentiality, they also prevent the client from querying encrypted data from the cloud.


This privacy versus data utilization dilemma may significantly degrade the benefits and usability of cloud systems. Therefore, it is vital to develop privacy enhancing technologies that can address this problem while retaining the practicality of the underlying cloud service. Searchable Symmetric Encryption (SSE) enables a client to encrypt data in such a way that they can later perform keyword searches on it. These encrypted queries are performed via “search tokens” over an encrypted index which represents the relationship between search token (keywords) and encrypted files.

Private-key storage outsourcing allows clients with either limited resources or limited expertise to store and distribute large amounts of symmetrically encrypted data at low cost. Since regular private-key encryption prevents one from searching over encrypted data, clients also lose the ability to selectively retrieve segments of their data. To address this, several techniques have been proposed for provisioning symmetric encryption with search capabilities the resulting construct is typically called searchable encryption.

2. LITERATURE SURVEY

Literature Survey-2.1

- Practical dynamic searchable encryption with small leakage, E. Stefanov, C. Papamanthou, and E. Shi in 2017.
- In this paper we revisit the DSSE problem. We propose the first DSSE scheme that achieves the best of both worlds, i.e., both small leakage and efficiency.

 (Enriching the Research)	Open Access Research Article
	Volume: 23 Issue: 07
	July, 2023

- In particular, our DSSE scheme leaks significantly less information than any other previous DSSE construction and supports both updates and searches in sublinear time in the worst case, maintaining at the same time a data structure of only linear size.

Literature Survey-2.2

- Dynamic searchable symmetric encryption, S. Kamara, C. Papamanthou, and T. Roeder in 2012.
- In this paper, Searchable Symmetric Encryption (SSE) enables a client to encrypt data in such a way that she can later perform keyword searches on it via “search tokens”.
- A prominent application of SSE is to enable privacy-preserving keyword searches on cloud based systems.

Literature Survey-2.3

- Practical techniques for searches on encrypted data, D. X. Song, D. Wagner, and A. Perrig in 2000.
- In this paper, we describe our cryptographic schemes for the problem of searching on encrypted data and provide proofs of security for the resulting crypto systems.
- They are provably secure, they provide query isolation for searches, they also support hidden queries.

Literature Survey-2.4

- Forward secure dynamic searchable symmetric encryption with efficient updates, K. S. Kim, M. Kim, D. Lee, J. H. Park, and W.-H. Kim in 2017.
- Searchable Symmetric Encryption (SSE) is an effective method to solve the problem of encrypted data retrieval, and helps to protect the users’ privacy. Based on symmetric encryption primitives, we propose a dynamic efficient forward privacy scheme DSSE.

DSSE uses pseudo-random permutation to realize forward privacy, and uses delete list to identify the final state of the same file added and deleted repeatedly, so as to realize the dynamic update of data.

3. EXISTING SYSTEM

Searchable Symmetric Encryption (SSE) enables a client to encrypt data in such a way that they can later perform keyword searches on it. These encrypted queries are performed via “search tokens” over an encrypted index which represents the relationship between search token (keywords) and encrypted files. A prominent application of SSE is to enable privacy-preserving keyword search on the cloud, where a data owner can outsource a collection of encrypted files and perform keyword searches on it without revealing the file and query contents. 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. Later, several Dynamic Searchable Symmetric Encryption (DSSE) schemes were proposed that permit the user to add and delete files after the system is set up. To the best of our knowledge, there is no single DSSE scheme that outperforms all the other alternatives in terms of all the aforementioned metrics: privacy (e.g., information leakage), performance (e.g., search, update delay), storage efficiency and functionality. In the following, we first provide an overview on DSSE

	Open Access Research Article
	Volume: 23 Issue: 07
	July, 2023

research and then, outline our research objectives and contributions toward addressing some of the limitations of the state-of-the-arts.

Disadvantages

In the existing work, the system leaks significant information for updates and it is not parallelizable. The existing several forward-private DSSE schemes achieving high efficiency in terms of asymptotic complexity and actual performance have been proposed.

4. PROPOSED SYSTEM

Although a number of DSSE schemes have been introduced the literature, most of them only provide a theoretical asymptotic analysis and, in some cases, merely a prototype implementation. The lack of experimental performance evaluations on real platforms poses a significant difficulty. We assessing the application and practicality of proposed DSSE schemes, as the impacts of security vulnerability, hidden computation costs, multi-round communication delay and storage blowup might be overlooked. Although several forward-secure DSSE schemes with an optimal asymptotic complexity have been proposed, they incur either very high delay due to public-key operations or significant storage blow-up at both client and server side and therefore, their ability to meet actual need of real systems in practice is still unclear.

Advantages


The system is more effective since a prominent application of SSE is to enable privacy-preserving keyword search on the cloud (e.g., Amazon S3), where a data owner can outsource a collection of encrypted files and perform keyword searches on it without revealing the file and query contents. The system is more secured since highly secure against File-Injection Attacks: IM-DSSE offers forward privacy or x4 for definition) which is an imperative security feature to mitigate the impact of practical file-injection attacks.

The purpose of the design phase is to arrange an answer of the matter such as by the necessity document. This part is that the opening moves in moving the matter domain to the answer domain. The design phase satisfies the requirements of the system. The design of a system is probably the foremost crucial issue warm heartedness the standard of the software package. It's a serious impact on the later part, notably testing and maintenance.

The output of this part is that the style of the document. This document is analogous to a blueprint of answer and is employed later throughout implementation, testing and maintenance. The design activity is commonly divided into 2 separate phases System Design and Detailed Design.

System Design conjointly referred to as top-ranking style aims to spot the modules that ought to be within the system, the specifications of those modules, and the way they move with one another to supply the specified results.

At the top of the system style all the main knowledge structures, file formats, output formats, and also the major modules within the system and their specifications square measure set. System design is that the method or art of process the design, components, modules, interfaces, and knowledge for a system to satisfy such as needs. Users will read it because the application of systems theory to development.

 (Enriching the Research)	Open Access Research Article
	Volume: 23 Issue: 07
	July, 2023

Detailed Design, the inner logic of every of the modules laid out in system design is determined. Throughout this part, the small print of the info of a module square measure sometimes laid out in a high-level style description language that is freelance of the target language within which the software package can eventually be enforced.

In system design the main target is on distinguishing the modules, whereas throughout careful style the main target is on

planning the logic for every of the modules.

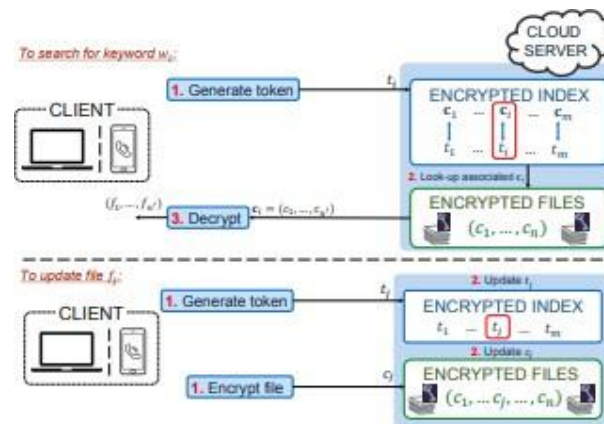


Fig 1 IM-DSSE framework for privacy file storage

The architecture of A Secure Searchable Encryption Framework for Privacy-Critical Cloud Storage Services typically consists of the following components:

1. **Client Application:** This is the interface that is used by users to interact with the cloud storage service. It enables users to upload, download, and search for files stored in the cloud. The client application is responsible for encrypting the files before they are uploaded to the cloud and decrypting them when they are downloaded.
2. **Cloud Storage Service:** This is the cloud-based storage platform where encrypted files are stored. The service should provide secure and reliable storage capabilities for the encrypted files.
3. **Searchable Encryption Module:** This module provides the ability to search for files stored in the cloud without compromising their confidentiality. It enables users to perform keyword searches on their files without having to decrypt them first.
4. **Key Management System:** This is responsible for generating and managing the encryption keys used to encrypt and decrypt the files. It is important to ensure that the keys are kept secure and that only authorized users have access to them.
5. **Access Control System:** This is responsible for controlling access to the encrypted files stored in the cloud. It should ensure that only authorized users are able to access the files.
6. **Audit Logging System:** This system maintains a log of all access attempts to the encrypted files stored in the cloud. It helps to detect and investigate any unauthorized access attempts.

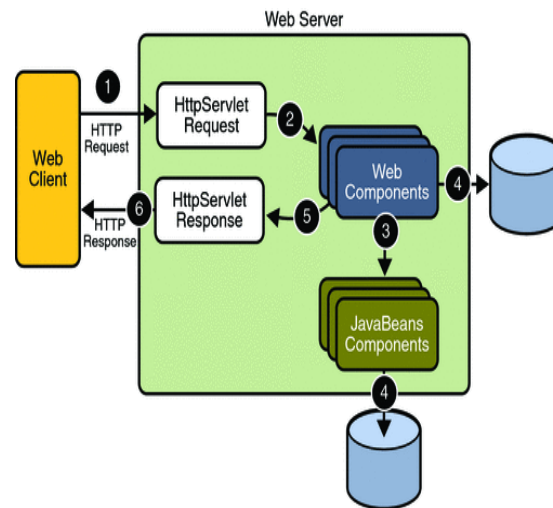


Fig .2 System Architecture

5. RESULTS



Fig 3 Main Page

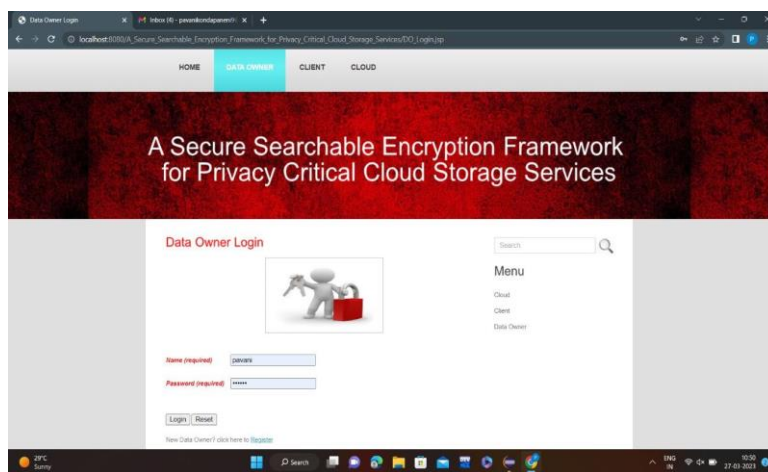


Fig 4 Data Owner Login

 <p>(Enriching the Research)</p>	Open Access Research Article
	Volume: 23 Issue: 07
	July, 2023

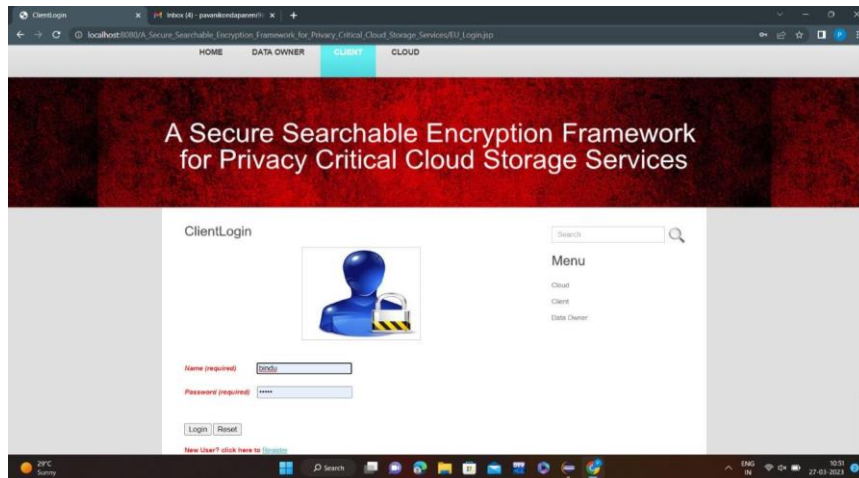


Fig 5 Client Login



Fig 6 Cloud Login



Fig 7 Data Owner Menu


 (Enriching the Research)	Open Access Research Article
	Volume: 23 Issue: 07
	July, 2023



Fig 8 Client Menu

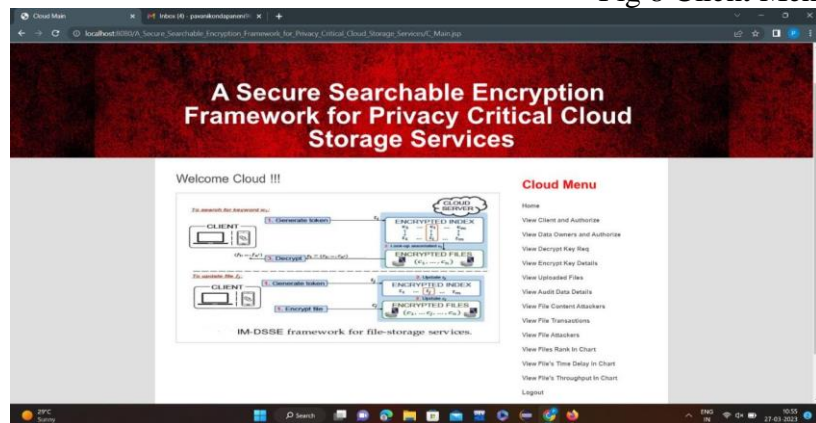



Fig 9 Cloud Menu

6. CONCLUSION

In this article, we presented IM-DSSE, a new DSSE framework which offers very high privacy, efficient updates, low search latency simultaneously. Our constructions rely on a simple yet efficient incidence matrix data structure in combination with two hash tables that allow efficient and secure search and update operations. Our framework offers various DSSE constructions, which are specifically designed to meet the needs of cloud infrastructure and personal usage in different applications and environments. All of our schemes in IM-DSSE framework are proven to be secure and achieve the highest privacy among their counterparts. We conducted a detailed experimental analysis to evaluate the performance of our schemes on real Amazon EC2 cloud systems. Our results showed the high practicality of our framework, even when deployed on mobile devices with large datasets. We have released the full-fledged implementation of our framework for public use and analysis.

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.

	Open Access Research Article
	Volume: 23 Issue: 07
	July, 2023

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. Krawczyk, 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 multi-keyword 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.
11. R. Bost, “Sophos – forward secure searchable encryption,” in Proc. 2016 ACM Conf. Comput. Commun. Security. ACM, 2016.
12. S. Kamara and T. Moataz, “Boolean searchable symmetric encryption with worst-case sub-linear complexity,” EUROCRYPT 2017, 2017.
13. D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Rosu, and M. Steiner, “Highly-scalable searchable symmetric encryption with support for boolean queries,” in Advances in Cryptology, CRYPTO 2013, ser. Lecture Notes in Comput. Sci., vol. 8042, 2013, pp. 353–373.
14. Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, “Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement,” IEEE Trans. Inform. Forensics Security, vol. 11, no. 12, pp. 2706–2716, 2016.
15. Q. Wang, M. He, M. Du, S. S. Chow, R. W. Lai, and Q. Zou, “Searchable encryption over feature-rich data,” IEEE Trans. Dependable Secure Computing, 2016.
16. Y. Zhang, J. Katz, and C. Papamanthou, “All your queries are belong to us: The power of file- injection attacks on searchable encryption,” in 25th USENIX Security ’16, Austin, TX, 2016, pp. 707–720.
17. A. A. Yavuz and J. Guajardo, “Dynamic searchable symmetric encryption with minimal leakage and efficient updates on commodity hardware,” in Int. Conf. Selected Areas in Cryptography. Springer, 2015, pp. 241–259.
18. P. Rizomiliotis and S. Gritzalis, “Oram based forward privacy preserving dynamic searchable symmetric encryption schemes,” in Proc. 2015 ACM Workshop Cloud Computing Security Workshop. ACM, 2015, pp. 65–76.