International Journal of Engineering Research and Science & Technology

DYNAMIC SEARCHABLE SYMMETRIC ENCRYPTION WITH STRONG SECURITY AND ROBUSTNESS

ROJARAMANI ADAPA[1], CHEVULA MAHESH[2], CHITNENI SRINILAYA[3], G. RAMAKRISHNA[4], THIRUNAGARI SRINIVAS[5]

[1]Assistant Professor, Dept. of CSE, Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Telangana, India

[2, 3, 4, 5] B.Tech, Dept. of CSE, Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Telangana, India

ABSTRACT— Dynamic Searchable Symmetric Encryption (DSSE) is a prospective technique in the field of cloud storage for secure search over encrypted data. A DSSE client can issue update queries to an honest-but-curious server for adding or deleting his cipher texts to or from the server and delegate keyword search over those cipher texts to the server. Numerous investigations focus on achieving strong security, like forward and-Type-I−-backward security, to reduce the information leakage of DSSE to the server as much as possible. However, the existing DSSE with such strong security cannot keep search correctness and stable security (or robustness, in short) if irrational queries are issued by the client, like duplicate add or delete queries and the delete queries for removing non-existed entries, to the server unintentionally. Hence, this work proposes two new DSSE schemes, named SR-DSSEa and SR-DSSEb, respectively. Both two schemes achieve forward and-Type-I−-backward security while keeping robustness when irrational queries are issued. In terms of performance, SR-DSSEa has more efficient communication costs and roundtrips than SR-DSSEb. In contrast, SR-DSSEb has a more efficient search performance than SR-DSSEa. Its search performance is close to the existing DSSE scheme with the same security but fails to achieve robustness.

Index Terms – Dynamic Searchable Symmetric Encryption (DSSE), searchable encryption, cloud storage security, forward security, backward security, robust searchable encryption, encrypted search, secure cloud storage, data privacy, dynamic updates, communication efficiency, cryptography.

I. INTRODUCTION

DYNAMIC Searchable Symmetric Encryption (DSSE) is a widely used technique for performing secure key word searches over ciphertexts that are constantly changing. In DSSE applications, all data of the client is encrypted and stored in remote environments like the cloud, which helps to maintain data confidentiality. DSSE enables the client to issue update queries to add or delete cipher texts to or from the cloud and delegate keyword search queries over his cipher texts to the cloud while maintaining keyword confidentiality [2]. Many software products, such as the Mistubishi Information and Communication System1 and the Crypteron security platform,2 have made extensive use of DSSE. Recently, numerous researchers have paid attention to developing DSSE with strong security to restrict the information leakage of DSSE as much as possible. To address these concerns, Stefanov et al. paid an apparent effort by defining two new kinds of security, named forward security and backward security [3]. The former restricts that information about the earlier queries' keywords is not leaked by any new update query, while the latter guarantees that an attacker cannot learn "too much" information about update queries issued between any two adjacent search queries. Following the seminal work, Bost et al. categorized backward security into three different types (from the strongest one to the weak est one), which is denoted as Type-I, Type-II, and Type-III, respectively, to restrict the information leakage in the degree from strong to weak [4]. To restrict the information leakage further, Zuo et al. proposed Type-I backward security, which is the strongest one so far as we know [5]. In brief, Type-I backward security requires that the information leakage caused by a search query contains which files match the query and when the related update queries. are issued. Note that in Type-I−-backward security, an attacker cannot distinguish add and delete queries. Compared with the Type-I−-backward security, the Type-I-backward security allows the search query to have an additional leakage of the time when the related add queries are issued. In Type-II backward security, leakage of the search query additionally consists of when the related add and delete queries are issued. Finally, compared with Type-II-backward security, the weaker Type-III-backward security also allows a search query to leak the relationships between the related add and delete queries, namely which add queries a delete query wants to remove. For the time

being, FB-DSSE is firstly proposed to achieve forward-and-Type-I−-backward security [5]. It uses a bitmap index to represent all possible files' identifiers. Each FB-DSSE ciphertext contains a keyword and an assigned bitmap index to denote which files pair with the keyword. When receiving a keyword search query, the server computes corresponding indexes at the beginning. Using these indexes, the server retrieves relevant cipher texts, aggregates those cipher texts into one by an addition homomorphism operation, and finally returns the client the aggregated cipher text. The returned cipher text contains an assigned bitmap index to denote all the files matching the search query. Later, three other DSSE schemes, named FBDSSE-CQ, SFBDSSE-CQ [9], and FBDSSE-RQ [10], respectively, were proposed to obtain the forward-and-Type-I−-backward security. In particular, both FBDSSE-CQ and SFBDSSE-CQ aim at achieving conjunctive keyword search, whereas FBDSSE-RQ aims at solving range keyword search. All the aforementioned works have achieved forward-and Type-I−-backward security. However, they fail to ensure stable search correctness and security (referred to as robustness for brevity) in case the client unintentionally issues irrational queries. The robustness of DSSE was first investigated by Xu et al. [8]. They demonstrated that a practical DSSE scheme must be robust since it is very hard to avoid the mistake caused by the careless client, like issuing duplicate add or delete queries, or removing non-existed entries by delete queries either. And they constructed a DSSE scheme named ROSE to obtain the robustness and the forward-and Type-III-backward security.

II. RELATED WORK

A. Forward and backward private DSSE for range queries

Due to its capabilities of searches and updates over the encrypted database, the dynamic searchable symmetric encryption (DSSE) has received considerable attention recently. To resist leakage abuse attacks, a secure DSSE scheme usually requires forward and backward privacy. However, the existing forward and backward private DSSE schemes either only support single keyword queries or require more interactions between the client and the server. In this article, we first give a new leakage function for range queries, which is more complicated than the one for single keyword queries. Furthermore, we propose a concrete forward and backward private DSSE scheme by using a refined binary tree data structure. Finally, the

detailed security analysis and extensive experiments demonstrate that our proposal is secure and efficient, respectively.

B. *Leakage inversion: Towards quantifying privacy in searchable encryption*

Searchable encryption (SE) provides cryptographic guarantees that a user can efficiently search over encrypted data while only disclosing patterns about the data, also known as leakage. Recently, the community has developed leakage-abuse attacks that shed light on what an attacker can infer about the underlying sensitive information using the aforementioned leakage. A glaring missing piece in this effort is the absence of a systematic and rigorous method that quantifies the privacy guarantees of SE. In this work, we put forth the notion of leakage inversion that quantifies privacy in SE. Our insight is that the leakage is a function and, thus, one can define its inverse which corresponds to the collection of databases that reveal structurally equivalent patterns to the original plaintext database. We call this collection of databases the reconstruction space and we rigorously study its properties that impact the privacy of an SE scheme such as the entropy of the reconstruction space and the distance of its members from the original plaintext database. Leakage inversion allows for a foundational algorithmic analysis of the privacy offered by SE and we demonstrate this by defining closed-form expressions and lower/upper bounds on the properties of the reconstruction space for both keyword- based and range-based databases. We use leakage inversion in three scenarios: (i) we quantify the impact that auxiliary information, a typical cryptanalytic assumption, has to the overall privacy, (ii) we quantify how privacy is affected in case of restricting range schemes to respond to a limited number of queries, and (iii) we study the efficiency vs. privacy trade-off offered by proposed padding defenses. We use real-world databases in all three scenarios and we draw theoretically-grounded new insights about the interplay between leakage attack.

C. *High recovery with fewer injections: Practical binary volumetric injection attacks against dynamic searchable encryption*

Searchable symmetric encryption enables private queries over an encrypted database, but it can also result in information leakages. Adversaries can exploit these leakages to launch injection attacks (Zhang et al., USENIX Security'16) to recover the underlying keywords from queries. The performance of the existing injection attacks is strongly dependent on the amount of

leaked information or injection. In this work, we propose two new injection attacks, namely BVA and BVMA, by leveraging a binary volumetric approach. We enable adversaries to inject fewer files than the existing volumetric attacks by using the known keywords and reveal the queries by observing the volume of the query results. Our attacks can thwart well-studied defenses (e.g., threshold countermeasure, padding) without exploiting the distribution of target queries and client databases. We evaluate the proposed attacks empirically in real-world datasets with practical queries. The results show that our attacks can obtain a high recovery rate (> 80%) in the best-case scenario and a roughly 60% recovery even under a large-scale dataset with a small number of injections.

D. The power of bamboo: On the post-compromise security for searchable symmetric encryption

Dynamic searchable symmetric encryption (DSSE) enables users to delegate the keyword search over dynamically updated encrypted databases to an honest-but-curious server without losing keyword privacy. This paper studies a new and practical security risk to DSSE, namely, secret key compromise (e.g., a user's secret key is leaked or stolen), which threatens all the security guarantees offered by existing DSSE schemes. To address this open problem, we introduce the notion of searchable encryption with key-update (SEKU) that provides users with the option of non-interactive key updates. We further define the notion of post-compromise secure with respect to leakage functions to study whether DSSE schemes can still provide data security after the client's secret key is compromised. We demonstrate that post-compromise security is achievable with a proposed protocol called "Bamboo". Interestingly, the leakage functions of Bamboo satisfy the requirements for both forward and backward security. We conduct a performance evaluation of Bamboo using a real-world dataset and compare its runtime efficiency with the existing forward-and-backward secure DSSE schemes. The result shows that Bamboo provides strong security with better or comparable performance.

III. PROPOSED METHODOLOGY

Dynamic Searchable Symmetric Encryption (DSSE) is a prospective technique in the field of cloud storage for secure search over encrypted data. A DSSE client can issue update queries to an honest-but-curious server for adding or deleting his ciphertexts to or from the server and delegate keyword search over those cipher texts to the server.

It uses a bitmap index and values to represent all files' identifiers. When receiving a keyword search query, the server computes corresponding indexes at the beginning. Using these indexes, the server retrieves relevant cipher texts, aggregates those cipher texts.

The returned cipher text contains an assigned bitmap index to denote all the files matching the search query.

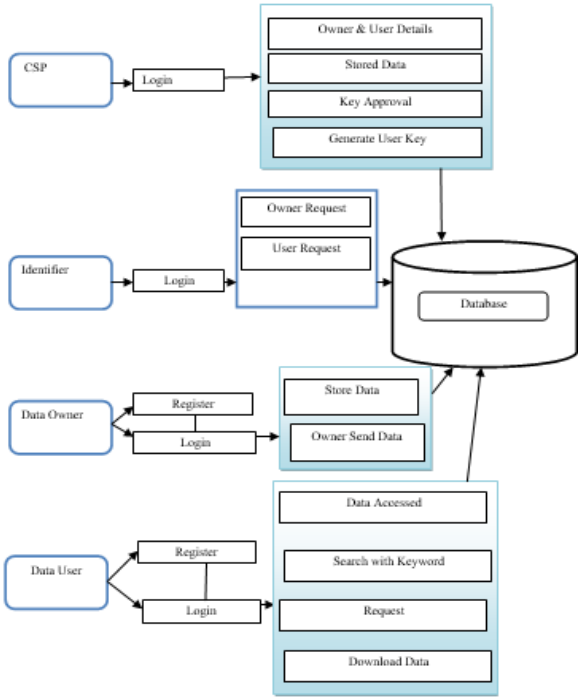
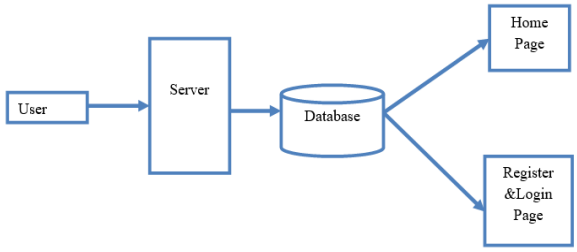


Fig. 1: System Overview

1. User Interface Design

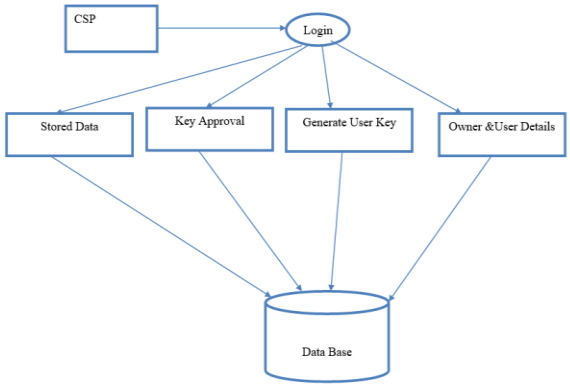
In this module we design the windows for the project. These windows are used for secure login for all users. To connect with server user must give their username and password then only they can able to connect the server.

If the user already exits directly can login into the server else user must register their details such as username, password and Email id, into the server. Server will create the account for the entire user to maintain upload and download rate. Name will be set as user id. Logging in is usually used to enter a specific page.



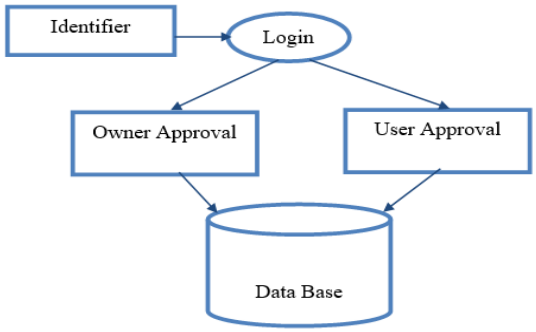
2. CSP

This is the first module the cloud services have a login with a mail id and password. The Cloud server has a stored information will be shows details. Cloud services have a key approval for data. There was a having information of a generate a user keys. It will have a owner and user details of a data.



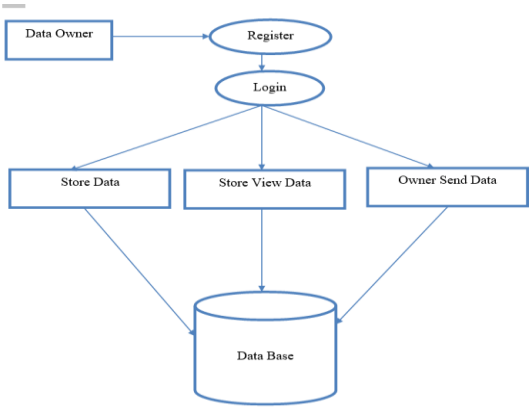
3. Identifier

This is the Second module of this project the identifier has a login with a mail id and password. A identifier has a approve a owner to login for permission. The user can also takes from permission for the identifier user approval to the database.



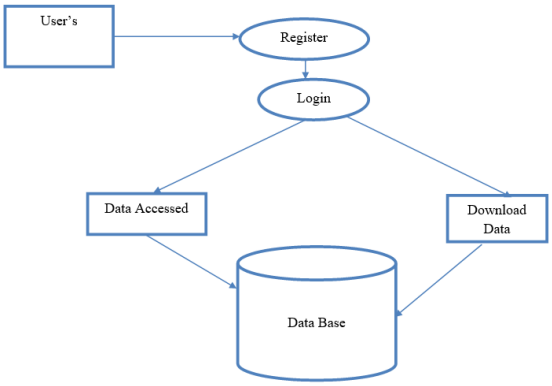
4. Data Owner

This is the third module of this project. In this project data owner has a register with all details and login with a user id and password. The owner has a store a file the file has an upload with a text file to the data. The data owner has a store a view data. The data owner has a owner send data.



5. Data User

This is the fourth module of this project. User's has a register with all details and then login with an email id and password. Users has a search the has a store we can search a data. The key response it will get a key responses. User has a key then it will have a decrypt a file download.



IV. EXPERIMENTAL RESULTS AND ANALYSIS

The experimental results show that the proposed dynamic AES encryption with blockchain-based key management significantly improves cloud data security. The use of unique AES keys for each file reduces the risk of data exposure, while blockchain provides secure and tamper-resistant key storage. The integration of Elliptic Curve Cryptography (ECC) further enhances secure key exchange and file sharing.

International Journal of Engineering Research and Science & Technology

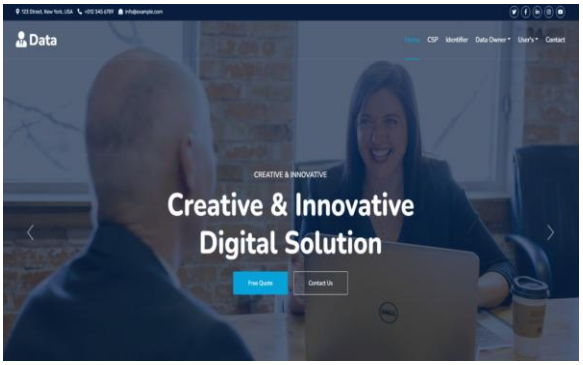


Fig. 2: Home Page

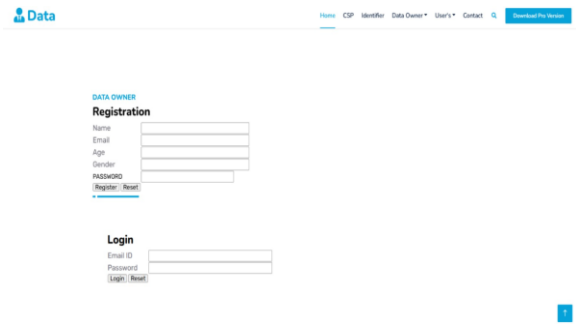


Fig. 3: Registration and Login

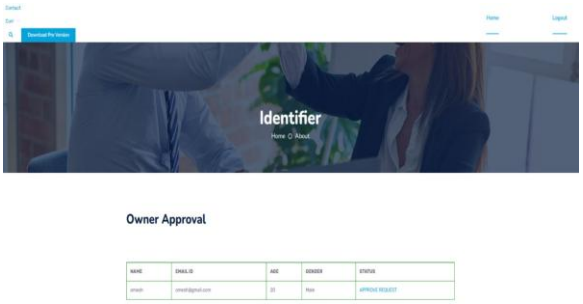


Fig. 4: Owner Approval

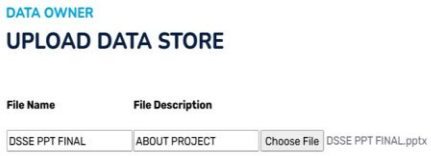


Fig. 5: Upload Data Store

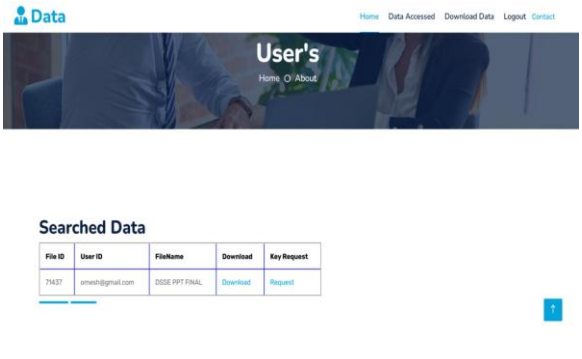


Fig. 6: Searched Data

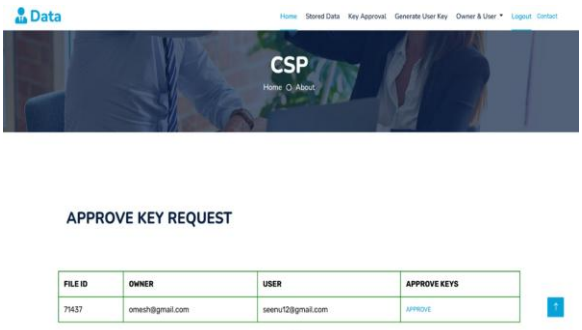


Fig. 7: Approved Key Request

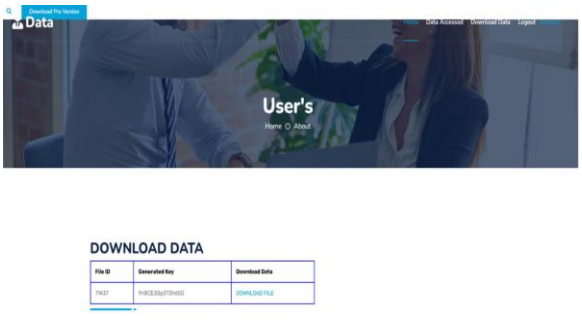


Fig. 8: Download Data

V. CONCLUSION

In this work, we identify the robustness problem existing in forward-and-Type-I−-backward private DSSE schemes. To solve this problem, the definition of Type-I− backward security is extended. With the new definitions, we constructed two novel robust

DSSE schemes, both of which achieves the security aim of forward and Type-I– backward privacy, i.e., SR-DSSEa and SR-DSSEb. The constructions of these two schemes leverage our newly proposed Bi-bitmap-index data structure and a Boolean circuit evaluation method. The experimental results show that SR-DSSEa is client-friendly and SR-DSSEb has higher search performance. The experiments show that SR-DSSEa and SR-DSSEb are not as performant as IM-DSSEI+II, thereby, may not be very suitable for some performance-intensive scenarios.

REFERENCES

[1] S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic searchable symmetric encryption," in Proc. ACM Conf. Comput. Commun. Secur., Oct. 2012, pp. 965–976.

[2] S. Lu, J. Zheng, Z. Cao, Y. Wang, and C. Gu, "A survey on cryptographic techniques for protecting big data security: Present and forthcoming," Sci. China Inf. Sci., vol. 65, no. 10, pp. 1–34, Oct. 2022.

[3] E. Stefanov, C. Papamanthou, and E. Shi, "Practical dynamic searchable encryption with small leakage," in Proc. Netw. Distrib. Syst. Secur. Symp., 2014, pp. 1–15.

[4] R. Bost, B. Minaud, and O. Ohrimenko, "Forward and backward pri vate searchable encryption from constrained cryptographic primitives," in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., Oct. 2017, pp. 1465–1482.

[5] C. Zuo, S. Sun, J. K. Liu, J. Shao, and J. Pieprzyk, "Dynamic searchable symmetric encryption with forward and stronger backward privacy," in Proc. ESORICS. Cham, Switzerland: Springer, 2019, pp. 283–303.

[6] T. Hoang, A. A. Yavuz, and J. Guajardo, "A secure searchable encryption framework for privacy-critical cloud storage services," IEEE Trans. Services Comput., vol. 14, no. 6, pp. 1675–1689, Nov. 2021.

[7] J. G. Chamani, D. Papadopoulos, C. Papamanthou, and R. Jalili, "New constructions for forward and backward private symmetric searchable encryption," in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., Oct. 2018, pp. 1038–1055.

[8] P. Xu et al., "ROSE: Robust searchable encryption with forward and backward

security," IEEE Trans. Inf. Forensics Security, vol. 17, pp. 1115–1130, 2022.

AUTHORS Profile



Mrs. ROJARAMANI ADAPA, received M.Tech (CSE) from JNTUH. She has teaching experience 20 years and has guided many undergraduate and post graduate projects. She has published about 10 research articles in peer reviewed journals and presented five research papers at various National Conferences. At present she is working as Asst. Prof. in CSE Department at Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Telangana, India.



Mr. CHEVULA MAHESH pursuing B.Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.



Ms. CHITNENI SRINILAYA pursuing B.Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.



Mr. G. RAMAKRISHNA pursuing B.Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.



Mr. THIRUNAGARI SRINIVAS pursuing B.Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022 2026.