DESIGNING DETECTION OF DATA LINEAGE FRAMEWORK IN ADVERSARIAL ENVIRONMENTS

PATAN NAYEEMULLAH KHAN PG Scholar, Dept of CSE, Quba College of Engineering & Technology, Venkatachalam, AP, India.

D.MURALI Assistant Professor, Dept of CSE, Quba College of Engineering & Technology, Venkatachalam, AP, India.

ABSTRACT

A growing number of customers are considering cloud computing as prospective hosts for their enormous datasets. Existing research recommends encrypting sensitive data before outsourcing and implementing Searchable Symmetric Encryption (SSE) to enable keyword-based searches over the ciphertexts because the cloud service provider (CSP) is outside the users' trusted domain. Designing a successful SSE system that concurrently supports sublinear search time, effective update and verification, and on-demand information retrieval is still a difficult issue.. To solve this, we provide the Verifiable Dynamic Encryption with Ranked Search (VDERS) method, which enables a user to securely and quickly perform top-K searches on a dynamic document collection and confirm the accuracy of the search results. To be more precise, we first offer a fundamental structure, VDERS0, that enables the insertion of verified documents in top-K searches by creating a ranking inverted index and a verifiable matrix. Subsequently, VDERS, a sophisticated architecture, is developed to facilitate document deletion even further while requiring less communication. Many tests on real datasets show how efficient and effective our VDERS system is.

1. INTRODUCTION

Cloud computing has received a lot of attention from the research and business industries as a potential computing paradigm. Users frequently outsource their big datasets to clouds and assign a cloud service provider (CSP) to take care of data storage and provide query services because of the advantages of low prices, flexibility, and scalability. Research currently available advises encrypting data before outsourcing due to security and privacy issues [1]. Yet, keyword-based searches over ciphertexts are a difficult problem due to data encryption. In a dynamic and malevolent cloud environment, this is considerably more difficult [2].

Let's think about the next circumstance. In the cloud, where each email is indexed by the sender's name and sorted by receipt date, Alice outsources her archived emails. For instance, the email received on April 2 has a higher rank than the email received on April 1 for a set of emails indexed by the term Bob. Alice uploads her documents and keywords to the cloud in encrypted form to protect their confidentiality. There could be hundreds of documents that match a given keyword, and the user would incur significant fees if they were all returned to them and decrypted. Alice may therefore wish to run a top-K search to find the most current emails..

Also, Alice might like to save money by just storing emails that she has received in the recent three months. For instance, Alice will delete all emails she received prior to February whenever May arrives. The chosen encryption method should adhere to the following specifications in the aforementioned application scenario: First, a ranked search. To find the papers that are most closely matched, the user is permitted to conduct a top-K search. (2) Active. Documents kept in the cloud can be updated (added and removed) by the user. (3) Validity. To conserve memory, the malicious CSP may destroy encrypted files that are not frequently accessed or it may falsify search results to trick the user..

A virus or worm can tamper with encrypted documents even if the CSP is truthful. As a result, the user should be able to confirm the accuracy of the search results. 4. Effectiveness. A set of encrypted documents may be efficiently searched, updated, and verified by the user..

Existing SSE schemes only partially meet the aforementioned requirements, despite the fact that Searchable Symmetric Encryption (SSE) enables users to access requested documents while protecting their anonymity. This work suggests a Verifiable Dynamic Encryption with Ranked Search (VDERS) method that enables the user to carry out updates and top-K searches on ciphertexts in a verifiable and effective manner in order to simultaneously satisfy all these features. The key concept behind our approach is to build a verifiable matrix to store the rating data and encrypt it using an RSA accumulator [3]. Furthermore, to enable effective top-K searches and updates, a ranking inverted index is created from a set of documents..To be more precise, we first offer a fundamental architecture, indicated by VDERS0, that allows for verified document insertion activities. Then, we offer a sophisticated design, abbreviated as VDERS, that not only enables effective deletion operations but also lowers communication costs without outsourcing the verifiable matrix. These are our primary contributions, in brief:

To enable dynamic and ranked searches in a cloud environment in an effective and verifiable manner, we suggest a VDERS scheme.To achieve effective top-K searches with support for verifiable updates, two architectures are offered.To demonstrate the success of our system, we carry out comprehensive experiments on real datasets and theoretically examine its performance and security.

2. PROBLEM STATEMENT

Since Song et al. initially suggested SSE, it has been the subject of extensive research. as a key contribution to SSE. A thorough security definition and two systems, SSE-1 and SSE-2, based on an inverted index, were developed by Curtmola et al. A user can quickly and securely update the encrypted outsourced data using Dynamic SSE (DSSE). With Verifiable SSE (VSSE), a user can confirm the accuracy of search results. A user can spot any dishonest server's cheating behaviour using the Universally Composable (UC)-secure VSSE technique developed by Kurosawa and colleagues. According to the quantity of matching query keywords, Cao et aldynamic .'s multi-keyword RSSE system ranked results using the secure KNN technique..

LIMITATION OF SYSTEM

Their plan's primary drawback is its inefficiency. The amount of documents increases linearly with the search time..

3. PROPOSED SYSTEM

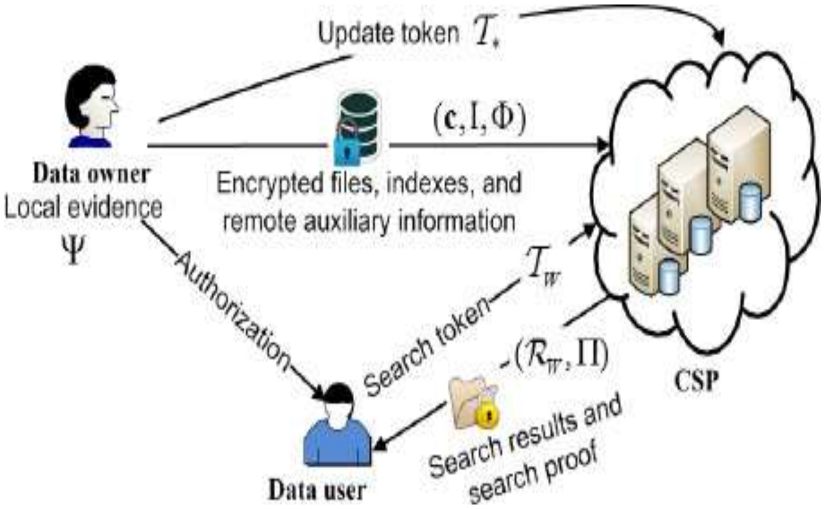
In order to achieve dynamic and ranked searches in a cloud environment in an effective and verifiable manner, we suggest a VDERS scheme. To obtain effective top-K searches with support for verifiable updates, two architectures are offered. To demonstrate the success of our system, we carry out comprehensive experiments on real datasets while theoretically analysing its performance and security..

ADVANTAGES OF PROPOSED SYSTEM

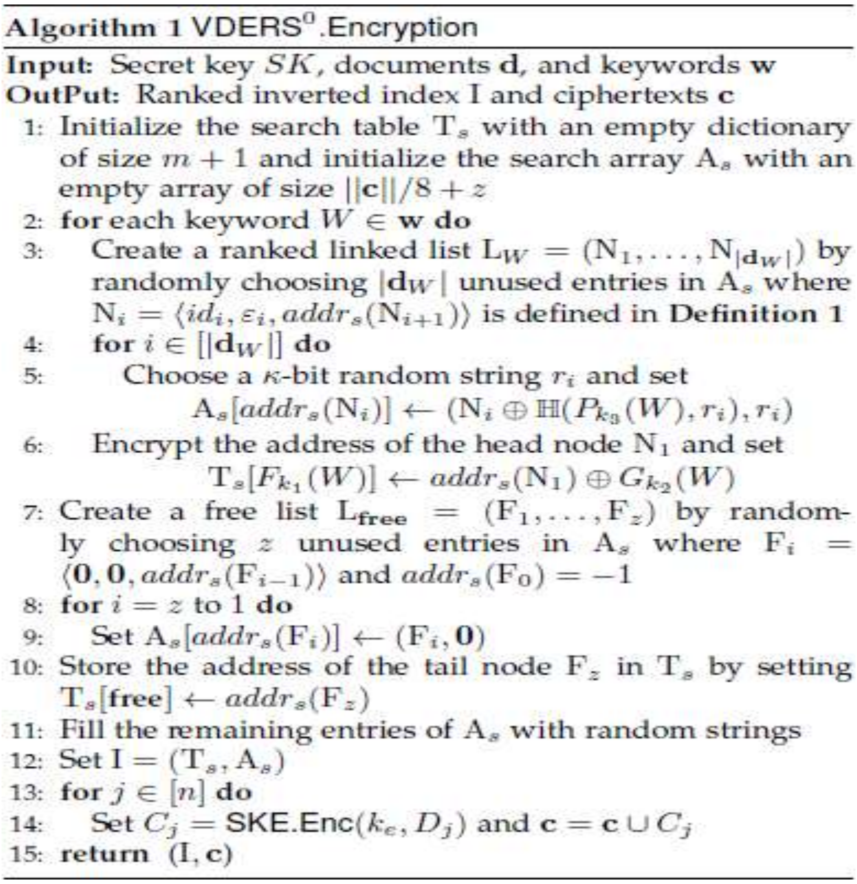
Their plan's primary drawback is its inefficiency. The amount of documents increases linearly with the length of the search. Designing SSE methods that facilitate dynamic update for cloud data is essential.

4. SYSTEM IMPLEMENTATION

As shown in Fig. 1, the system is composed of three distinct parties: the CSP, the data owner, and the data consumer. Data storage and query services are provided by the CSP using cloud platforms that pool hard and soft resources. Cryphertexts are first created by the data owner for a document collection d. She then constructs a secure index I for quick searches using the keywords w derived from d, and produces local evidence and distant auxiliary information for verifiable queries. She can update ciphertexts with an update token T after uploading (c, I,) to the cloud and retrieve documents whenever she needs them with a search token T in a verifiable manner.After confirming the accuracy of the search results, the data owner recovers document contents after getting the search results and a search proof (RW,) from the CSP. The data owner can also assign authorised data users the responsibility for searching, updating, and verifying the data. In this work, we refer to both the data owner and the data user together as "users".

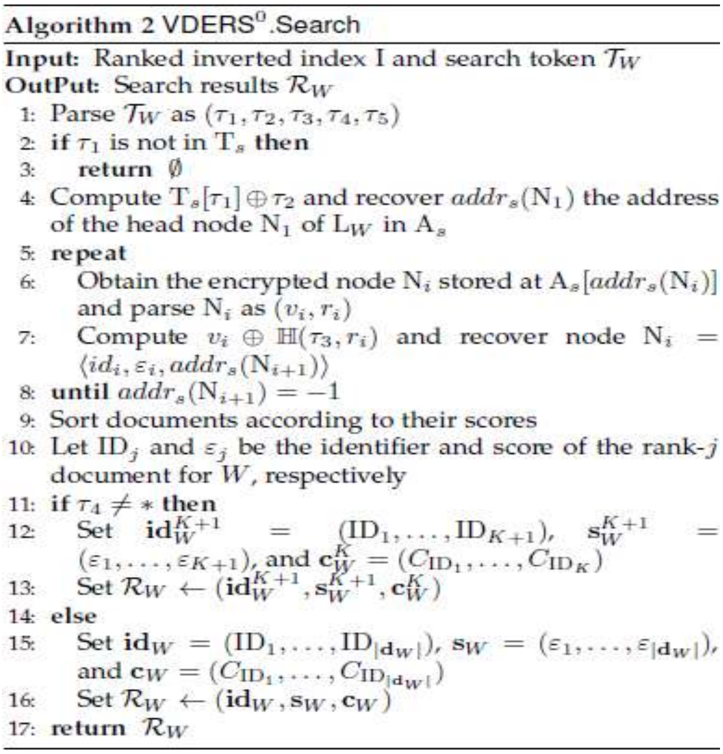

Update token $\mathcal{T}_*$
Data owner
Local evidence
Ψ
$(\mathbf{c}, \mathrm{I}, \Phi)$
Encrypted files, indexes, and remote auxiliary information
Authorization
Search token
$\mathcal{T}_W$
$(\mathcal{R}_W, \Pi)$
CSP
Search results and search proof
Data user

5. IMPLEMENTATIONALGORITHM



Algorithm 1 VDERS^0.Encryption

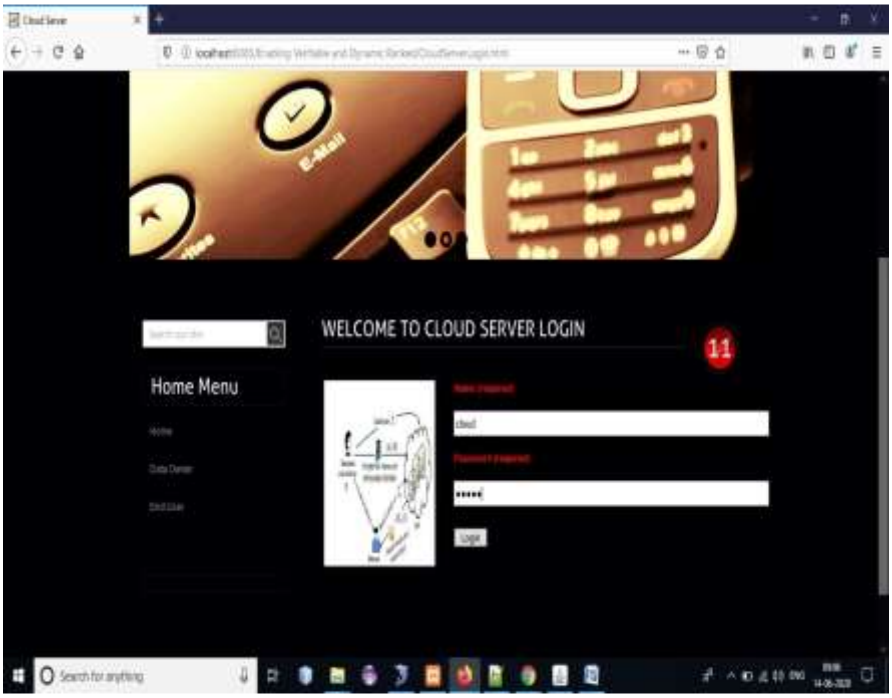
```
Input: Secret key $SK$, documents $\mathbf{d}$, and keywords $\mathbf{w}$
OutPut: Ranked inverted index I and ciphertexts $\mathbf{c}$
Initialize the search table $\mathrm{T}_s$ with an empty dictionary of size $m + 1$ and initialize the search array $\mathrm{A}_s$ with an empty array of size $||\mathbf{c}||/8 + z$
for each keyword $W \in \mathbf{w}$ do
    Create a ranked linked list $\mathrm{L}_W = (\mathrm{N}_1, \ldots, \mathrm{N}_{|\mathbf{d}_W|})$ by randomly choosing $|\mathbf{d}_W|$ unused entries in $\mathrm{A}_s$ where $\mathrm{N}_i = \langle id_i, \varepsilon_i, addr_s(\mathrm{N}_{i+1})\rangle$ is defined in Definition 1
    for $i \in [|\mathbf{d}_W|]$ do
        Choose a $\kappa$-bit random string $r_i$ and set
            $\mathrm{A}_s[addr_s(\mathrm{N}_i)] \leftarrow (\mathrm{N}_i \oplus \mathbb{H}(P_{k_3}(W), r_i), r_i)$
    Encrypt the address of the head node $\mathrm{N}_1$ and set
        $\mathrm{T}_s[F_{k_1}(W)] \leftarrow addr_s(\mathrm{N}_1) \oplus G_{k_2}(W)$
Create a free list $\mathrm{L}_{\mathbf{free}} = (\mathrm{F}_1, \ldots, \mathrm{F}_z)$ by randomly choosing $z$ unused entries in $\mathrm{A}_s$ where $\mathrm{F}_i = \langle \mathbf{0}, \mathbf{0}, addr_s(\mathrm{F}_{i-1})\rangle$ and $addr_s(\mathrm{F}_0) = -1$
for $i = z$ to 1 do
    Set $\mathrm{A}_s[addr_s(\mathrm{F}_i)] \leftarrow (\mathrm{F}_i, \mathbf{0})$
Store the address of the tail node $\mathrm{F}_z$ in $\mathrm{T}_s$ by setting $\mathrm{T}_s[\mathbf{free}] \leftarrow addr_s(\mathrm{F}_z)$
Fill the remaining entries of $\mathrm{A}_s$ with random strings
Set $\mathrm{I} = (\mathrm{T}_s, \mathrm{A}_s)$
for $j \in [n]$ do
    Set $C_j = \mathsf{SKE.Enc}(k_e, D_j)$ and $\mathbf{c} = \mathbf{c} \cup C_j$
return $(\mathrm{I}, \mathbf{c})$
```



Algorithm 2 VDERS^0.Search

```
Input: Ranked inverted index I and search token $\mathcal{T}_W$
OutPut: Search results $\mathcal{R}_W$
Parse $\mathcal{T}_W$ as $(\tau_1, \tau_2, \tau_3, \tau_4, \tau_5)$
if $\tau_1$ is not in $\mathrm{T}_s$ then
    return $\emptyset$
Compute $\mathrm{T}_s[\tau_1] \oplus \tau_2$ and recover $addr_s(\mathrm{N}_1)$ the address of the head node $\mathrm{N}_1$ of $\mathrm{L}_W$ in $\mathrm{A}_s$
repeat
    Obtain the encrypted node $\mathrm{N}_i$ stored at $\mathrm{A}_s[addr_s(\mathrm{N}_i)]$ and parse $\mathrm{N}_i$ as $(v_i, r_i)$
    Compute $v_i \oplus \mathbb{H}(\tau_3, r_i)$ and recover node $\mathrm{N}_i = \langle id_i, \varepsilon_i, addr_s(\mathrm{N}_{i+1})\rangle$
until $addr_s(\mathrm{N}_{i+1}) = -1$
Sort documents according to their scores
Let $\mathrm{ID}_j$ and $\varepsilon_j$ be the identifier and score of the rank-$j$ document for $W$, respectively
if $\tau_4 \neq *$ then
    Set $\mathbf{id}_W^{K+1} = (\mathrm{ID}_1, \ldots, \mathrm{ID}_{K+1})$, $\mathbf{s}_W^{K+1} = (\varepsilon_1, \ldots, \varepsilon_{K+1})$, and $\mathbf{c}_W^K = (C_{\mathrm{ID}_1}, \ldots, C_{\mathrm{ID}_K})$
    Set $\mathcal{R}_W \leftarrow (\mathbf{id}_W^{K+1}, \mathbf{s}_W^{K+1}, \mathbf{c}_W^K)$
else
    Set $\mathbf{id}_W = (\mathrm{ID}_1, \ldots, \mathrm{ID}_{|\mathbf{d}_W|})$, $\mathbf{s}_W = (\varepsilon_1, \ldots, \varepsilon_{|\mathbf{d}_W|})$, and $\mathbf{c}_W = (C_{\mathrm{ID}_1}, \ldots, C_{\mathrm{ID}_{|\mathbf{d}_W|}})$
    Set $\mathcal{R}_W \leftarrow (\mathbf{id}_W, \mathbf{s}_W, \mathbf{c}_W)$
return $\mathcal{R}_W$
```

6. OUTPUTSCREENS



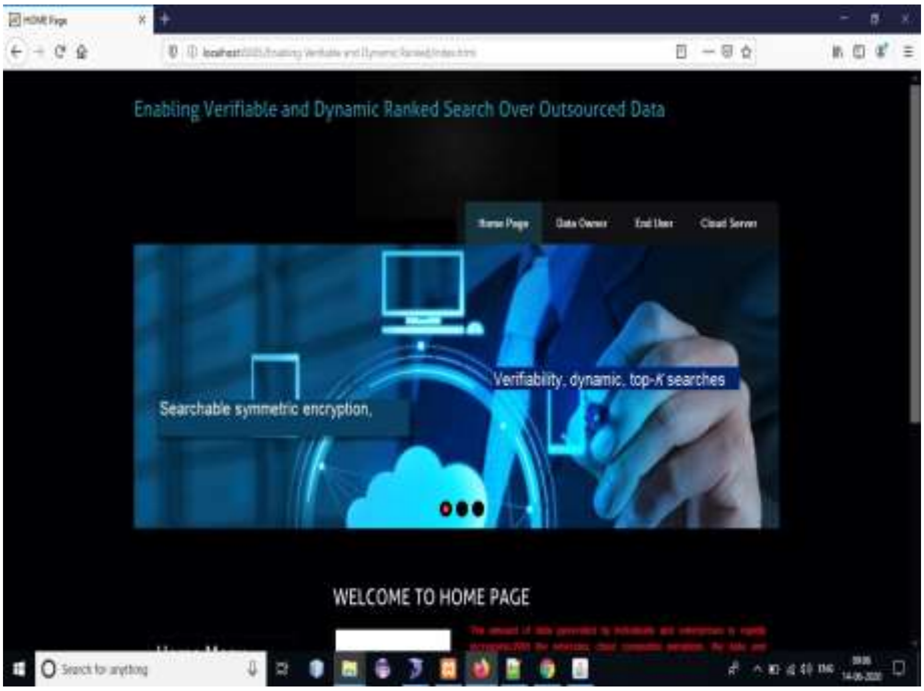
WELCOME TO CLOUD SERVER LOGIN

Name (required)

cloud

Password (required)

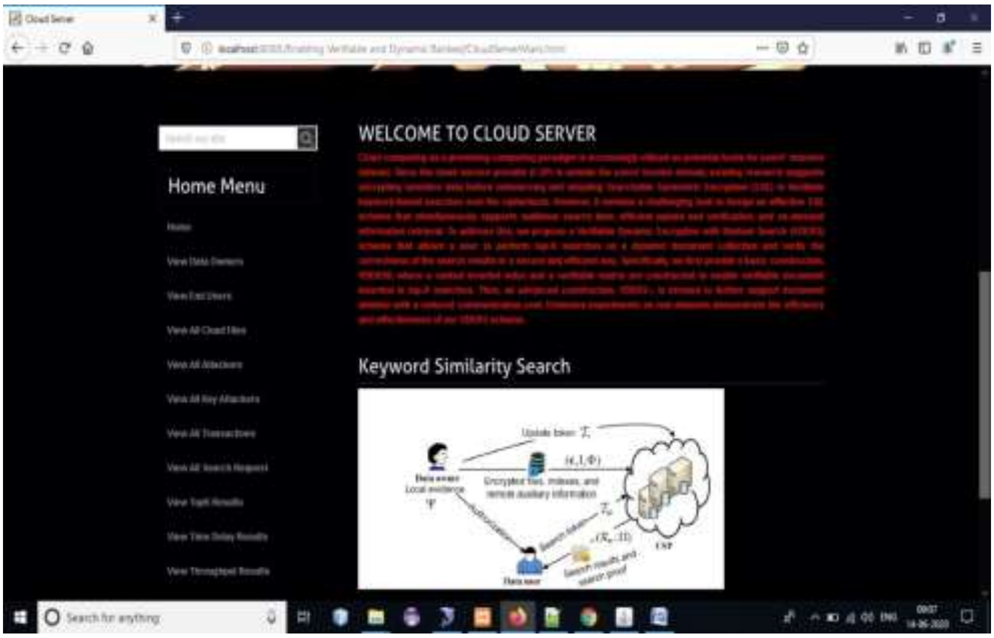
•••••



Enabling Verifiable and Dynamic Ranked Search Over Outsourced Data

WELCOME TO HOME PAGE

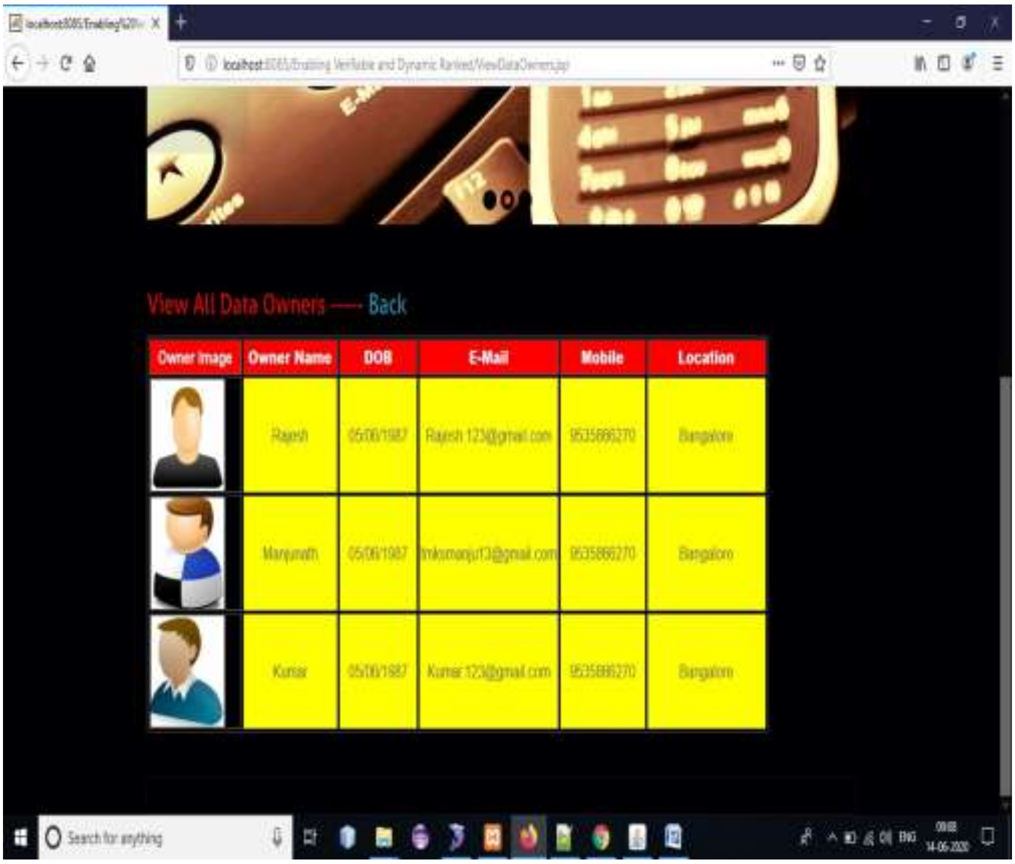
The amount of data generated by individuals and enterprises is rapidly increasing.With the emerging cloud computing paradigm, the data and



WELCOME TO CLOUD SERVER

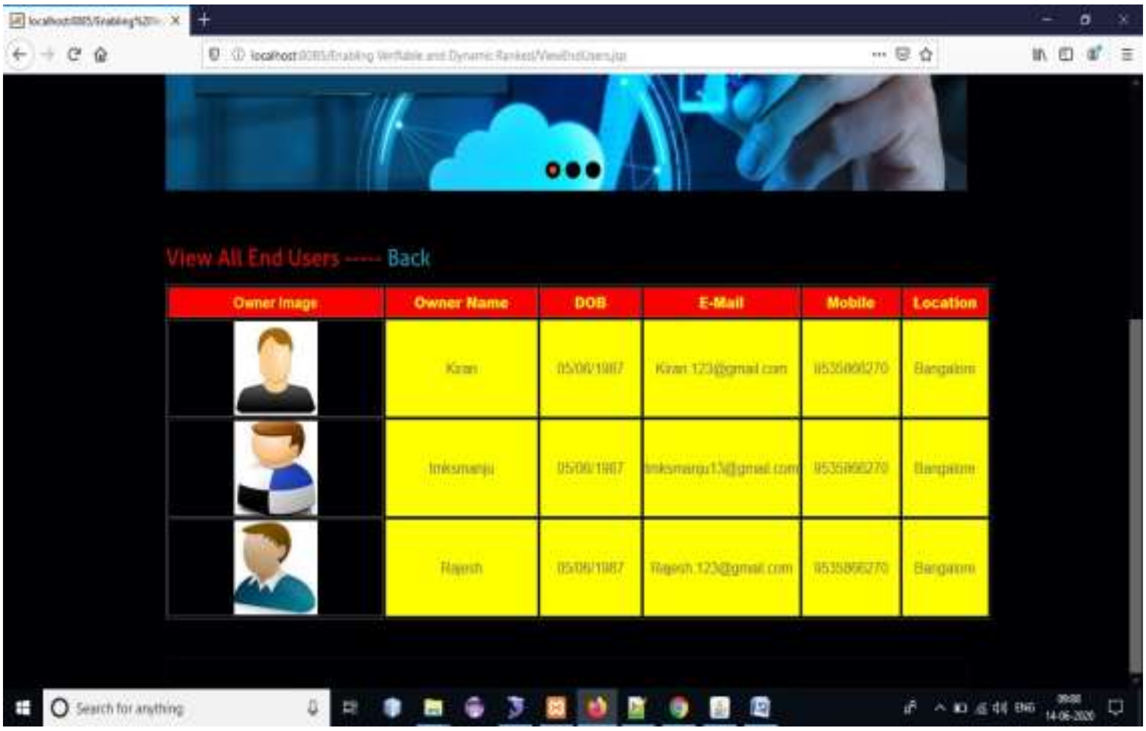
Cloud computing as a promising computing paradigm is increasingly utilized as potential hosts for users' massive dataset. Since the cloud service provider (CSP) is outside the users' trusted domain, existing research suggests encrypting sensitive data before outsourcing and adopting Searchable Symmetric Encryption (SSE) to facilitate keyword-based searches over the ciphertexts. However, it remains a challenging task to design an effective SSE scheme that simultaneously supports sublinear search time, efficient update and verification, and on-demand information retrieval. To address this, we propose a Verifiable Dynamic Encryption with Ranked Search (VDERS) scheme that allows a user to perform top-k searches on a dynamic document collection and verify the correctness of the search results in a secure and efficient way. Specifically, we first provide a basic construction, VDERS0, where a ranked inverted index and a verifiable matrix are constructed to enable verifiable document insertion in top-k searches. Then, an advanced construction, VDERS+, is devised to further support document deletion with a reduced communication cost. Extensive experiments on real datasets demonstrate the efficiency and effectiveness of our VDERS scheme.

Keyword Similarity Search


localhost:8085/Enabling%20V
localhost:8085/Enabling Verifiable and Dynamic Ranked/ViewDataOwners.jsp
Search for anything

View All Data Owners —— Back

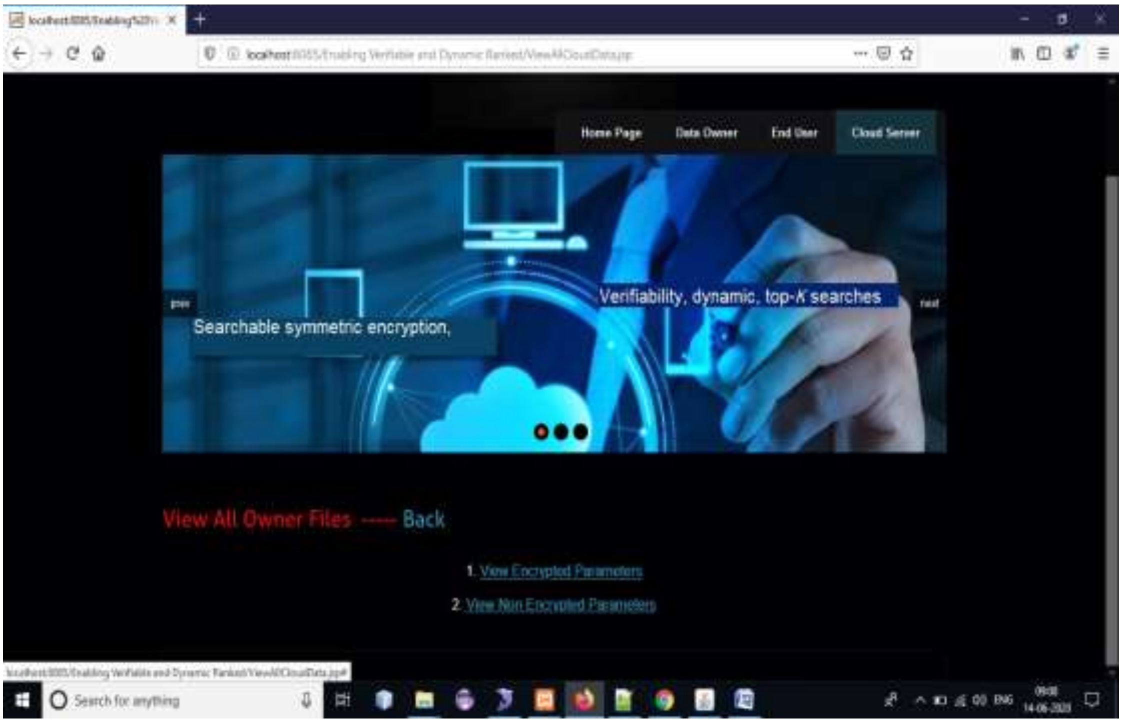
Owner Image	Owner Name	DOB	E-Mail	Mobile	Location
	Rajesh	05/06/1987	Rajesh.123@gmail.com	9535866270	Bangalore
	Manjunath	05/06/1987	tmksmanju13@gmail.com	9535866270	Bangalore
	Kumar	05/06/1987	Kumar.123@gmail.com	9535866270	Bangalore



localhost:8085/Enabling Verifiable and Dynamic Ranked/ViewEndUsers.jsp
Search for anything

View All End Users ----- Back

Owner Image	Owner Name	DOB	E-Mail	Mobile	Location
	Kiran	05/06/1987	Kiran.123@gmail.com	9535866270	Bangalore
	tmksmanju	05/06/1987	tmksmanju13@gmail.com	9535866270	Bangalore
	Rajesh	05/06/1987	Rajesh.123@gmail.com	9535866270	Bangalore

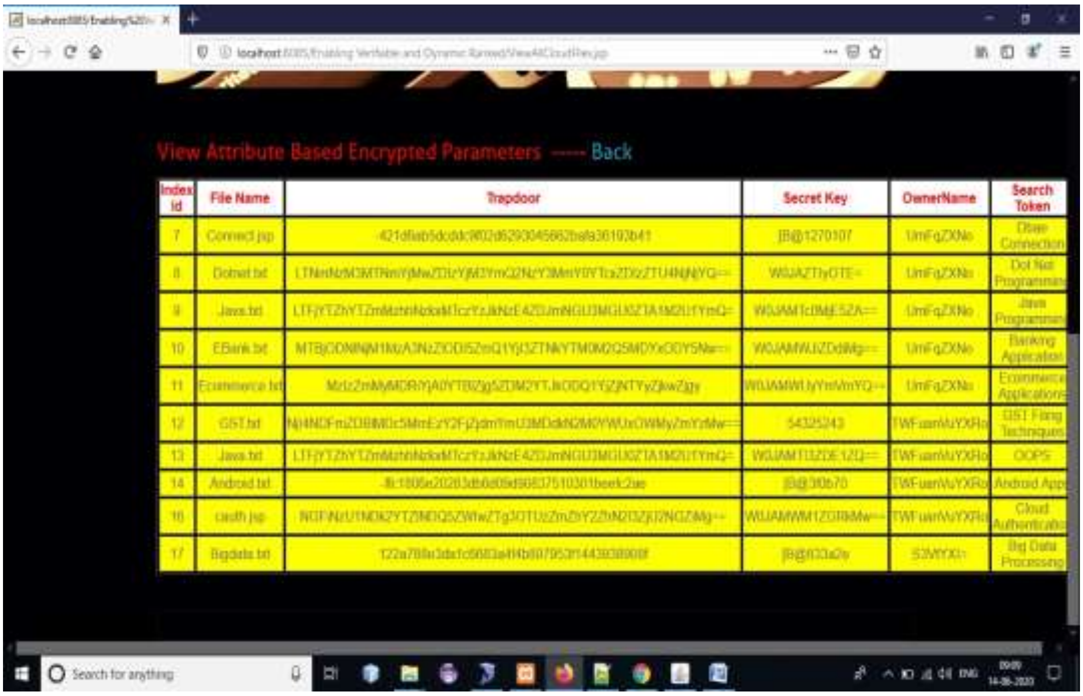


localhost:8085/Enabling Verifiable and Dynamic Ranked/ViewAllCloudData.jsp
Home Page
Data Owner
End User
Cloud Server
prev
Searchable symmetric encryption,
Verifiability, dynamic, top-K searches
next
Search for anything

View All Owner Files ----- Back

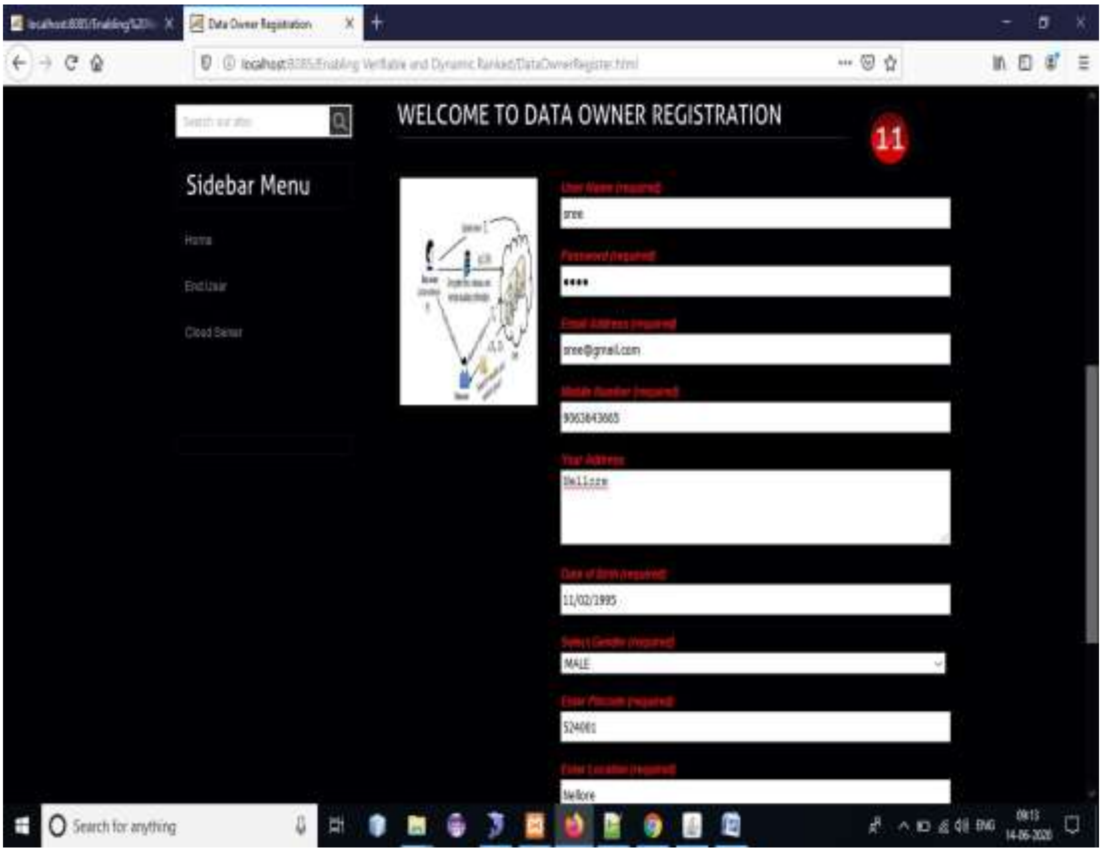
1. View Encrypted Parameters

2. View Non Encrypted Parameters



View Attribute Based Encrypted Parameters ----- Back

Index Id	File Name	Trapdoor	Secret Key	OwnerName	Search Token
7	Connect.jsp	421dfab5dcddc9f02d6293045662bafa36193b41	[B@1270107	UmFqZXNo	Dbase Connection
8	Dotnet.txt	LTNmNzM3MTNmYjMwZDIzYjM3YmQ2NzY3MmY0YTcxZDIzZTU4NjNjYQ==	W0JAZTIyOTE=	UmFqZXNo	Dot Net Programming
9	Java.txt	LTFjYTZhYTZmMzhhNzkxMTczYzJkNzE4ZDJmNGU3MGU5ZTA1M2UtYmQ=	W0JAMTc0MjE5ZA==	UmFqZXNo	Java Programming
10	EBank.txt	MTBjODNlNjM1MzA3NzZiODI5ZmQ1YjI3ZTNkYTM0M2Q5MDYxODY5Nw==	W0JAMWJiZDdiMg==	UmFqZXNo	Banking Application
11	Ecommerce.txt	MzIzZmMyMDRiYjA0YTBlZjg5ZDM2YTJkODQ1YjZjNTYyZjkwZjgy	W0JAMWI1YWVmYQ==	UmFqZXNo	Ecommerce Applications
12	GST.txt	NjI4NDFmZDBlM0c5MmEzY2FjZjdmYmU3MDdkN2M0YWUxOWMyZmYzMw==	54325243	TWFuamVuYXRo	GST Filing Techniques
13	Java.txt	LTFjYTZhYTZmMzhhNzkxMTczYzJkNzE4ZDJmNGU3MGU5ZTA1M2UtYmQ=	W0JAMTU2ZDE1ZQ==	TWFuamVuYXRo	OOPS
14	Android.txt	-8c1806e20283db6809d9683751030 1beek2ae	[B@3f0b70	TWFuamVuYXRo	Android App
16	cauth.jsp	NGFiNzU1NDk2YTZlMDQ5ZWIwZTg3OTUzZmZlY2ZhN2I3ZjI2NGZiMg==	W0JAMWM1ZGRkMw==	TWFuamVuYXRo	Cloud Authentication
17	Bigdata.txt	122a788e3dafc6683a4f4b897953f1443838908f	[B@f033a2e	S3VyYXJ=	Big Data Processing



WELCOME TO DATA OWNER REGISTRATION

User Name (required)
sree

Password (required)
••••

Email Address (required)
sree@gmail.com

Mobile Number (required)
9063843665

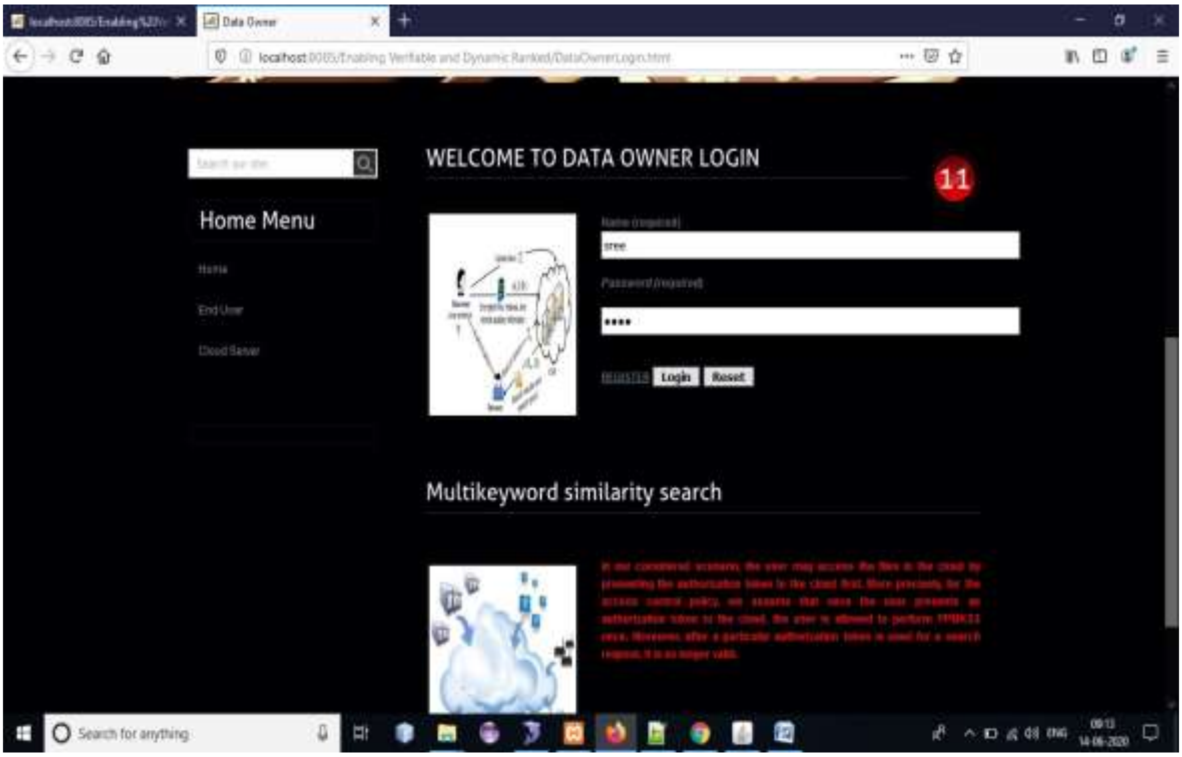
Your Address
Nellore

Date of Birth (required)
11/02/1995

Select Gender (required)
MALE

Enter Pincode (required)
524001

Enter Location (required)
Nellore


localhost:8085/Enabling Verifiable and Dynamic Ranked/DataOwnerLogin.html
Home Menu
Home
End User
Cloud Server
11
REGISTER
Login
Reset

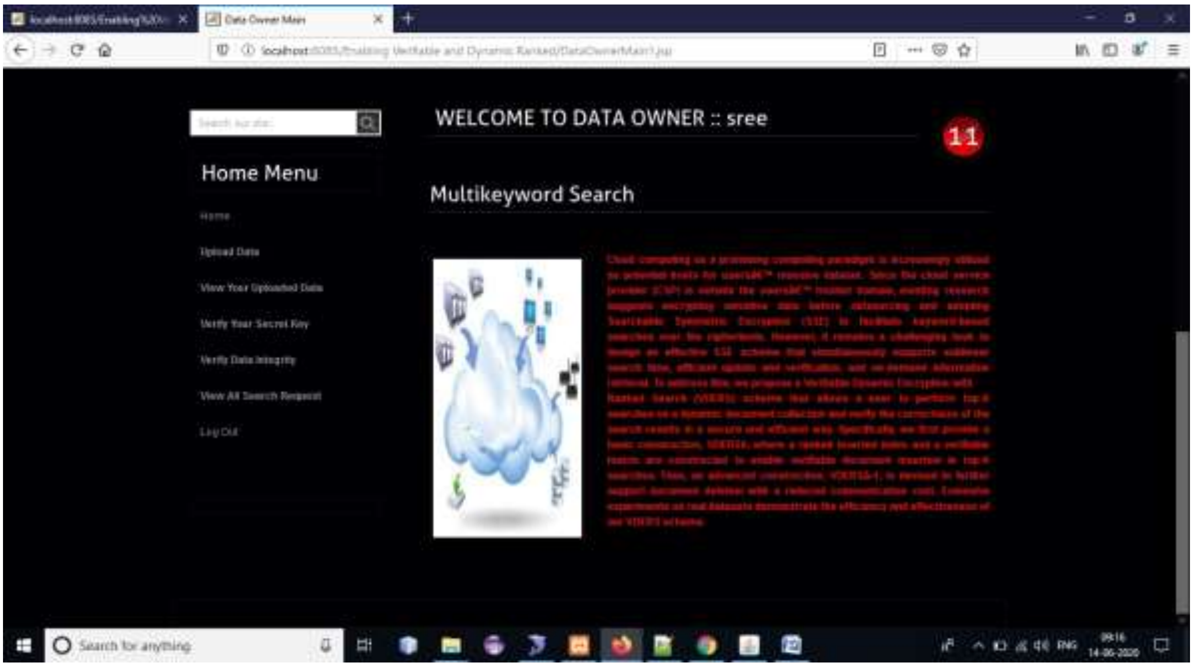
WELCOME TO DATA OWNER LOGIN

Name (required)
sree

Password (required)
****

Multikeyword similarity search

In our considered scenario, the user may access the files in the cloud by presenting the authorization token to the cloud first. More precisely, for the access control policy, we assume that once the user presents an authorization token to the cloud, the user is allowed to perform TPMKSS once. However, after a particular authorization token is used for a search request, it is no longer valid.



WELCOME TO DATA OWNER :: sree

Multikeyword Search

Cloud computing as a promising computing paradigm is increasingly utilized as potential hosts for usersâ€™ massive dataset. Since the cloud service provider (CSP) is outside the usersâ€™ trusted domain, existing research suggests encrypting sensitive data before outsourcing and adopting Searchable Symmetric Encryption (SSE) to facilitate keyword-based searches over the ciphertexts. However, it remains a challenging task to design an effective SSE scheme that simultaneously supports sublinear search time, efficient update and verification, and on-demand information retrieval. To address this, we propose a Verifiable Dynamic Encryption with Ranked Search (VDERS) scheme that allows a user to perform top-K searches on a dynamic document collection and verify the correctness of the search results in a secure and efficient way. Specifically, we first provide a basic construction, VDERSI, where a ranked inverted index and a verifiable matrix are constructed to enable verifiable document insertion in top-K searches. Then, an advanced construction, VDERSII, is devised to further support document deletion with a reduced communication cost. Extensive experiments on real datasets demonstrate the efficiency and effectiveness of our VDERS scheme.

7. CONCLUSIONS AND DISCUSSIONS

In this research, we build a VDERS technique to simultaneously allow on-demand information retrieval in a cloud computing environment, efficient updating and verification, and sublinear search time. Results from experiments show how useful our method is for confirming the accuracy of top-K searches on a dynamic document collection. We will endeavour to create a forward secure VDERS scheme in the next work that prevents keyword information about newly uploaded documents from being leaked during the update phase..

REFERENCES

[1] G. S. Poh, J.-J. Chin,W.-C. Yau, K.-K. R. Choo, and M. S. Mohamad, “Searchable symmetric encryption: designs and challenges,” ACM Computing Surveys (CSUR), 2017.

[2] Q. Liu, X. Nie, X. Liu, T. Peng, and J.Wu, “Verifiable ranked search over dynamic encrypted data in cloud computing,” in Proc. Of IWQoS, 2017.

[3] J. Camenisch and A. Lysyanskaya, “Dynamic accumulators and application to efficient revocation of anonymous credentials,” in Proc. of CRYPTO, 2002.

[4] D. X. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in Proc. of IEEE S&P, 2000.

[5] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: improved definitions and efficient constructions,” in Proc. of the ACM CCS, 2006.

[6] S. Kamara, C. Papamanthou, and T. Roeder, “Dynamic searchable symmetric encryption,” in Proc. of ACM CCS, 2012.

[7] S. Kamara and C. Papamanthou, “Parallel and dynamic searchable symmetric encryption,” in Proc. of FC, 2013.

[8] D. Cash, J. Jaeger, S. Jarecki, C. Jutla, H. Krawczyk, M. Rosu, and M. Steiner, “Dynamic searchable encryption in very-large databases: data structures and implementation,” in Proc. of NDSS, 2014.

[9] M. Naveed, M. Prabhakaran, and C. A. Gunter, “Dynamic searchable encryption via blind storage,” in Proc. of S&P, 2014.

[10] Y. Zhang, J. Katz, and C. Papamanthou, “All your queries are belong to us: The power of file-injection attacks on searchable encryption,” in Proc. of USENIX Security Symposium, 2016.

[11] E. Stefanov, C. Papamanthou, and E. Shi, “Practical dynamic searchable encryption with small leakage,” in Proc. of NDSS, 2014.

[12] R. Bost, οφος: Forward secure searchable encryption, in Proc. Of CCS, 2016.

[13] X. Song, C. Dong, D. Yuan, Q. Xu and M. Zhao, “Forward private searchable symmetric encryption with optimized I/O efficiency,” in IEEE Transactions on Dependable and Secure Computing, 2018.

[14] K. Kurosawa and Y. Ohtaki, “Uc-secure searchable symmetric encryption,” in Proc. of FC, 2012.

[15] Soleimanian, Azam,and S. Khazaei, “Publicly verifiable searchable symmetric encryption based on efficient cryptographic components,” in Designs, Codes and Cryptography, 2019.

[16] K. Kurosawa and Y. Ohtaki, “How to update documents verifiably in searchable symmetric encryption,” in Proc. of CNS, 2013.

[17] W. Sun, X. Liu, W. Lou, Y. T. Hou, and H. Li, “Catch you if you lie to me: efficient verifiable conjunctive keyword search over large dynamic encrypted cloud data,” in Proc. of IEEE INFOCOM, 2015.

[18] S. Jiang, X. Zhu, L. Guo, and J. Liu, “Publicly verifiable Boolean query over outsourced encrypted data,” in IEEE Transactions on Cloud Computing, 2017.

[19] J. Zhu, Q. Li, C. Wang, X. Yuan, Q. Wang and K. Ren, “Enabling generic, verifiable, and secure data search in cloud services,” in IEEE Transactions on Parallel and Distributed Systems, 2018.

[20] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, “Privacy-preserving multi-keyword ranked search over encrypted cloud data,” IEEE Transactions on Parallel and Distributed Systems, 2014.

[21] W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis, “Secure kNN computation on encrypted databases,” in Proc, of ACM SIGMOD, 2009.

[22] 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 Transactions on Parallel and Distributed Systems, 2014.

[23] C. Chen, X. Zhu, P. Shen, J. Hu, S. Guo, Z. Tari, and A. Y. Zomaya, “An efficient privacy-preserving ranked keyword search method,” IEEE Transactions on Parallel and Distributed Systems, 2016. [24] W. Zhang, Y. Lin, S. Xiao, J. Wu, and S. Zhou, “Privacy preserving ranked multi-keyword search for multiple data owners in cloud computing,” IEEE Transactions on Computers, 2016.

[25] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Hua,“Enabling personalized search over encrypted outsourced data with efficiency improvement,” IEEE transactions on parallel and distributed systems, 2016.

[26] C. Guo, X. Chen, Y. Jie, Z. Fu, M. Li, and B. Feng, “Dynamic multi-phrase ranked search over encrypted data with symmetric searchable encryption,” in IEEE Transactions on Services Computing, 2017.

[27] F. Kerschbaum, “Frequency-hiding order-preserving encryption,” in Proc. of CCS, 2015.

[28] D. S. Roche, D. Apon, S. G. Choi, and A. Yerukhimovich, “Pope: Partial order preserving encoding,” in Proc. of CCS, 2016.

[29] R. Gennaro, S. Halevi, and T. Rabin, “Secure hash-and-sign signatures without the random oracle,” in Stern J. (eds) Advances in Cryptology, 1999.

[30] N. Bari´c, B. Pfitzmann, “Collision-free accumulators and fail-stop signatrue schemes without trees,” in Fumy W. (eds) Advances in Cryptology, 1997.