ISSN: 2278-4632
Vol-10 Issue-6 No. 7 June 2020

Dynamic Ranked Search to Verifiable Over Outsourced Data in Cloud

[1]*D. Saritha Reddy* [2]*M. Lokesh*

[1]*Assistant Professor, Dept. of Master of Computer Applications, Narayana Engineering college, Gudur.*

[2]*PG Scholar, Dept. of Master of Computer Applications, Narayana Engineering College, Gudur.*

Abstract—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.

Keywords: Searchable Symmetric Encryption, Verifiability, Dynamic, Top-K Searches.

1. INTRODUCTION

As a promising computing paradigm, cloud computing has drawn great attention from both research and industry communities. Because of the benefits of low costs, flexibility, and scalability, it has become a prevalent trend for users to outsource their massive datasets to clouds and delegate a cloud service provider (CSP) to manage data storage and offer query services. Due to security and privacy concerns, existing research suggests encrypting data before outsourcing [1]. However, data encryption makes keyword-based searches over ciphertexts a challenging problem. This is even harder for efficient top-K searches in a dynamic and malicious cloud environment [2].

Let us consider the following scenario. Alice outsources archived emails to the cloud, where each email is indexed by the sender's name and ranked in descending order of the receipt date. For example, for a set of emails indexed by keyword Bob, the email received on April 2 has a higher rank than the email received on April 1. To keep keyword and document

ISSN: 2278-4632
Vol-10 Issue-6 No. 7 June 2020

contents secret, Alice uploads them in the encrypted forms to the cloud. There could be hundreds of documents matching a specific keyword, and the consumed costs will be extensive if all the matched documents are returned to and decrypted by the user. Therefore, Alice may want to perform a top-K search to retrieve the most recent emails. Moreover, Alice may want to store only the emails received in the last three months for monetary saving. For example, when entering May, Alice will delete all emails received before February.

In the above application scenario, the adopted encryption scheme should meet the following requirements: (1) Ranked search. The user is allowed to perform a top-K search to retrieve the best-matched documents. (2) Dynamic. The user is able to update (add and delete) documents stored in the cloud. (3) Verifiability. The malicious CSP may delete encrypted documents not commonly used to save memory space, or it may forge the search results to deceive the user. Even if the CSP is honest, a virus or worm may tamper with encrypted documents. Therefore, the user should have the ability to verify the correctness of the search results. (4) Efficiency. The user can efficiently perform searches, updates, and verifications on a set of encrypted documents.

Although Searchable Symmetric Encryption (SSE)allows a user to retrieve desired documents in a privacy-preserving way, existing SSE schemes only partially address the above requirements. To simultaneously satisfy all these properties, this paper proposes a Verifiable Dynamic Encryption with Ranked Search (VDERS) scheme that allows the user to perform updates and top-K searches on ciphertexts in a verifiable and efficient way. Our main idea is to construct a verifiable matrix to record the ranking information and encode it with RSA accumulator [3]. Furthermore, a ranked inverted index is built from a collection of documents to facilitate efficient top-K searches and updates. Specifically, we first provide a basic construction, denoted by VDERS0, which enables verifiable document insertion operations. Then, we provide an advanced construction, denoted by VDERS⋆, which not only can support efficient deletion operations, but also can reduce communication costs without outsourcing the verifiable matrix.

Our main contributions are summarized as follows:

• We propose a VDERS scheme to achieve dynamic and ranked searches in a cloud environment in an efficient and verifiable way.

• Two constructions are provided to achieve efficient top-K searches with support for verifiable updates.

• We theoretically analyze the security and performance of our scheme and conduct extensive experiments on real datasets to validate its effectiveness.

ISSN: 2278-4632
Vol-10 Issue-6 No. 7 June 2020

2. RELATED WORK

Our work focuses on verifiable ranked queries on dynamic encrypted data. SSE that allows an untrusted server to perform keyword-based searches on ciphertexts can partially address our requirements.

SSE has been widely researched since it was first proposed by Song et al. [4]. As a seminal work in SSE, Curtmola et al. [5] provided a rigorous security definition and constructed two schemes, SSE-1 and SSE-2, based on an inverted index. Compared to SSE-1, SSE-2 is more secure, and it has been proven to be secure against adaptive chosen keyword attacks (CKA2). SSE-1 is secure against nonadaptive chosen-keyword attacks (CKA1), but yields an optimal (sublinear) search time O(r), where r is the number of documents that contain the query keyword. However, neither SSE-1 nor SSE-2 has properties of dynamism, verifiability, and ranked search.

Dynamic SSE (DSSE) allows a user to update the encrypted outsourced data in an efficient and secure way. Kamaraetal.[6] constructed a DSSE scheme based on an extended inverted index. Their subsequent work [7] extended it to a parallel search setting by using a red-black tree. Cash et al. [8] constructed a DSSE scheme optimized for super large data sets, but their scheme supports only efficient document insertion. Naveedetal.[9] put forward a DSSE scheme in which a server worked as a blind storage to decrease leakage at the cost of multiple rounds of interaction. The above schemes are proven to be CKA2-secure, but leak keyword information about the newly added documents. With this leakage, the attackers can reveal the content of a past query by injecting new documents in a dataset [10]. To mitigate such an attack, Stefanovetal.[11] proposed the first DSSE scheme with forward privacy, but their scheme suffers from inefficiency. Since then, forward privacy has been the main motivation of recent DSSE schemes [12], [13].

Verifiable SSE (VSSE) allows a user to verify the correctness of search results. Kurosawa et al. [14] constructed a Universally Composable (UC)-secure VSSE scheme, in which a user can detect any malicious server's cheating behavior. WhileUC-securityisstrongerthanCKA2-security, their construction requires a linear search time. Soleimanianetal.[15] presented a public VSSE scheme, which delegates a third party to accomplish the verification, but fails to support dynamic operations. The subsequent work of Kurosawa et.al [16] extends VSSE to a dynamic environment.

Ranked SSE (RSSE) allows a user to rank the search results based on different evaluation functions. Cao et al. [20] proposed a dynamic multi-keyword RSSE scheme which utilized the secure KNN technique to rank results according to the number of matched query

ISSN: 2278-4632
Vol-10 Issue-6 No. 7 June 2020

keywords. The main limitation of their scheme is inefficiency, where the search time increases linearly with the number of documents. Inspired by their work, many multi-keyword VRSSE schemes were put forward. Sunetal. proposed a verifiable RSSE scheme which ranked search results based on cosine similarity while improving search efficiency by an MDB tree structure. Chen et al. proposed a RSSE scheme based on a hierarchical clustering index. Their scheme supports update and verification, but requires the client to re-encrypt search indexes once a document is added into a dataset. Zhang et al. developed a RSSE scheme in a multi-owner model, where the search time grew linearly with the number of queried keywords. Fu et al. leveraged a user interest model to assign weight to query keywords according to the user's search history. Guo et al. put forward a dynamic multi-phrase SSE scheme that allowed the user to rank results locally based on TF-IDF scores. However, it is hard for existing RSSE schemes to simultaneously support sublinear search time and efficient updates and verification.

3. PROPOSED WORK

A) System Model

The system consists of three different parties: the CSP, the data owner, and the data user, as illustrated in Fig. 1.

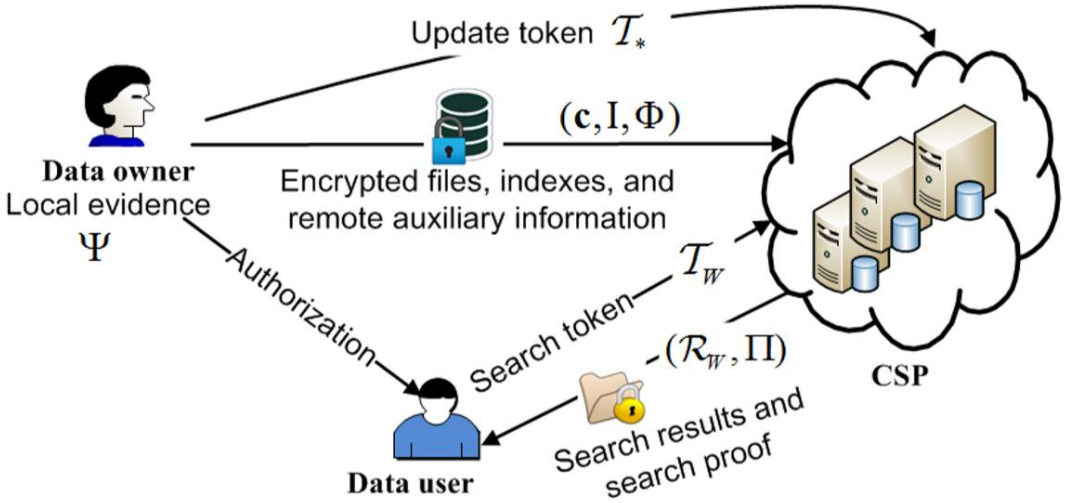


Fig. 1. System Model

This system model consists and implements the following modules:

Data Owner:

- In this module, Data owner can register and login to the system and he upload the file into untrusted cloud server.
- Here, he can view the uploaded file details and checks whether file data is safe or not.
- He can also view the users requests and allow them to search the data.

ISSN: 2278-4632
Vol-10 Issue-6 No. 7 June 2020

Data User:

- In this module, data user/end user can register and login to the system and he search the file if he has search permission.
- If he is not having search permission, he/she can request the search permission to data owner
- Once the data owner provides permission then he searches file and send request to cloud server for decrypt and download file.
- After getting permission from cloud server he/she can decrypt and download files.

Cloud:

- In this module, cloud provider login to the system and view all users' details, authorize them.
- Cloud provider can view the files uploaded to the system and he check the file decrypt and download requests and responses.
- He can also generate the various reports.

B) Basic VDERS Scheme

The foundational technologies in our VDERS scheme are RSA accumulator, the ranked inverted index, and the verifiable matrix (defined in Section 4.3). Our main idea is first building a ranked inverted index to achieve sublinear search time in top-K queries, and then verifying the correctness of search results based on RSA accumulator and the verifiable matrix. Specially, the search index is constructed as a ranked inverted index. Unlike the inverted index [5] in which each node contains only information about document identifier and the location of the next node in the search array, our ranked inverted index incorporates ranking information in each node to enable top-K queries in cloud computing. Furthermore, inspired by the work in [16], our VDERS scheme also employs RSA accumulator for dynamic verification. The main difference from their work is that our scheme constructs a verifiable matrix V to record the document ranking information for each keyword, and calculates the remote auxiliary information Φ and a local evidence Ψ by encoding the elements of V into RSA accumulator.

C) Advanced VDERS Scheme

Our basic VDERS construction allows top-K queries in a dynamic and verifiable way, but falls short in the following aspects: (1) Inefficiency in deletion operation. It flags the deleted

ISSN: 2278-4632
Vol-10 Issue-6 No. 7 June 2020

document but keeps corresponding nodes in the search array As. Therefore, it has no real meaning to delete a document and limits the number of the documents that can be added. (2) Inefficiency in verification. It requires to transmit the verifiable matrix V to the CSP for the generation of search proofs. The complexity of V is O (nmκ), which may be unacceptable for a large-scale dataset.

D) Algorithms

The proposed scheme consists of the following algorithms:

1. *GenKey (1κ):* It takes a security parameter κ as input and outputs public parameters params and a secret key SK.
2. *Encryption (SK, d, w):* It takes a secret keySK , a sequence of documents d, and a sequence of keywords was inputs, and outputs a ranked inverted index I and a sequence of ciphertexts c.
3. *AccGen (params, SK, d, w, c):* It takes public parameters params, a secret key SK, a sequence of documents d, a sequence of keywords w, and a sequence of ciphertexts c as inputs. It outputs remote auxiliary information Φ and a local evidence Ψ.
4. *SrcToken (SK, W):* It takes a secret key SK and a keyword W ∈ was inputs to generate a search token TW.
5. *Search (I, TW):* It takes a search token TW and an index I as inputs to output search resultsRW.
6. *GenProof (params, c, Φ, TW, RW):* It takes public parameters params, a sequence of ciphertexts c, remote auxiliary information Φ, a search token TW, and search results RW as inputs to generate a search proof Π.
7. *Verify (params, SK, Ψ, TW, RW, Π):* It takes public parameters params, a secret key SK, a local evidence Ψ, a search token TW, search results RW, and a search proof Π as inputs. It outputs 1 if verification is successful, and outputs 0 otherwise. If the output is 1, for each ciphertext in RW, it takes a secret key SK as input to recover plaintext.
8. *UpdToken (params, SK, D, Ψ):* It takes public parameters params, a secret key SK, a document D, and a local evidence Ψ as inputs to generate an update token T∗ and a new local evidence Ψ′, where∗∈ {add, del}.
9. *Update (I, c, Φ, T∗):* It takes a ranked inverted index I, a ciphertext collection c, remote auxiliary information Φ, and an update token T∗ as inputs. It generates a new

ISSN: 2278-4632
Vol-10 Issue-6 No. 7 June 2020

ranked inverted index I′, a new ciphertext collection c′, and new remote auxiliary information Φ′.

4. EXPERIMENT RESULTS & DISCUSION

In this section, we provide the proposed scheme comparative analysis results.

We compared our VDERS constructions with BASELINE1 and BASELINE2 in terms of the communication cost and the execution time in the initial phase, search phase, and update phase. Since BASELINE2 is unverifiable, it lacks algorithms AccGen, GenProof, and Verify.

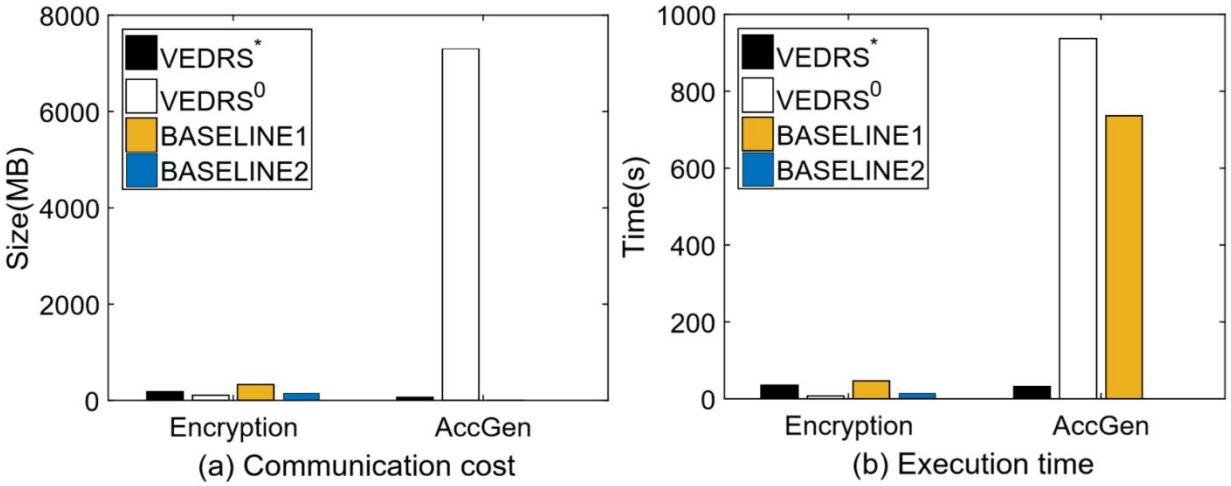
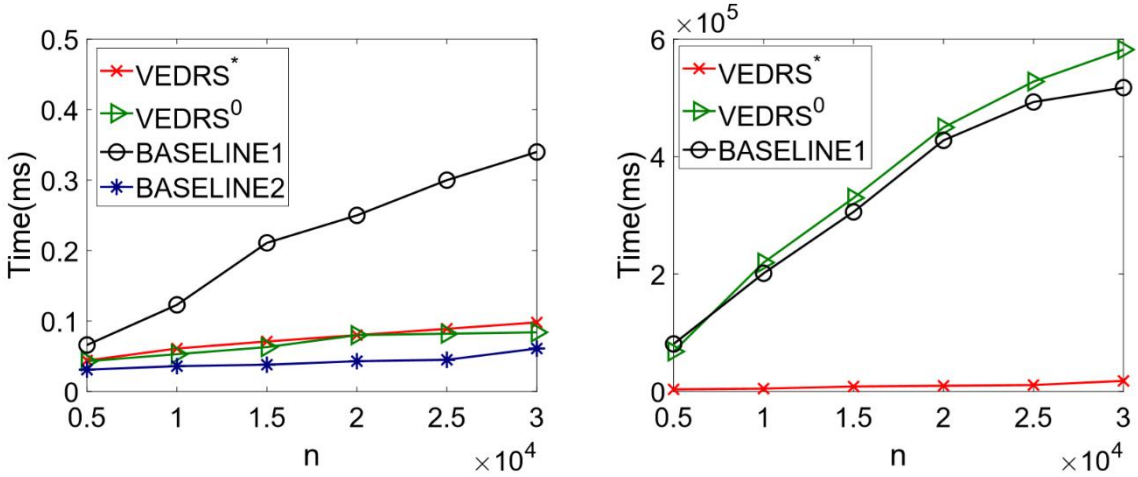


Fig. 2. Comparison in initial phase

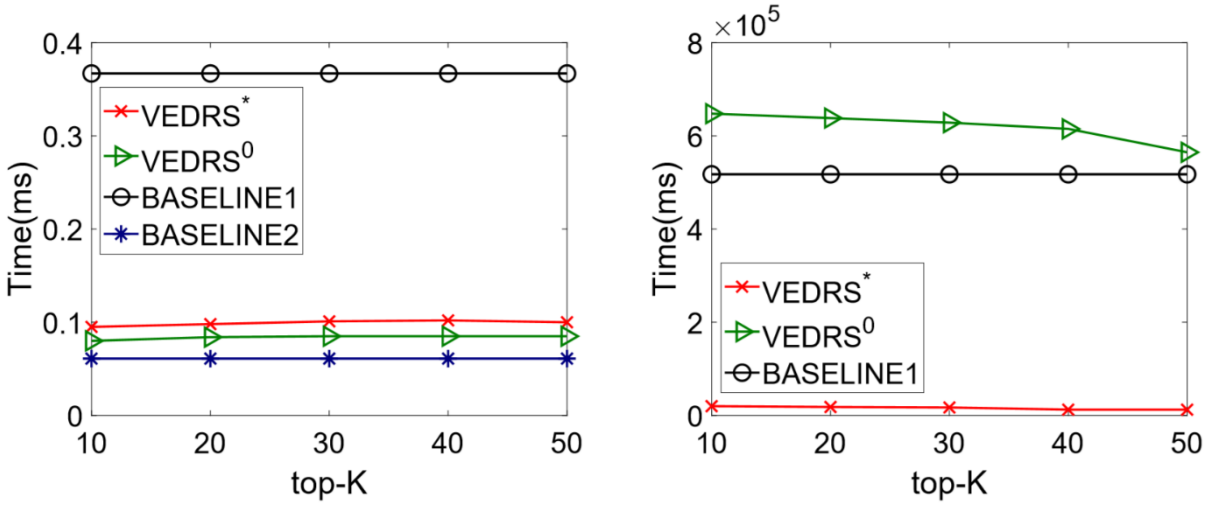
Fig. 2 illustrates the comparison results in the initial phase. For the Encryption algorithm, BASELINE1 encrypts a string of length n for every keyword, and thus incurs the most computation and communication costs. VDERS0 and VDERS⋆ encrypt a ranked inverted index in the same way.

Fig. 3 shows the execution time at the server side in the search phase.



ISSN: 2278-4632
Vol-10 Issue-6 No. 7 June 2020

(a) Search (b) GenProof



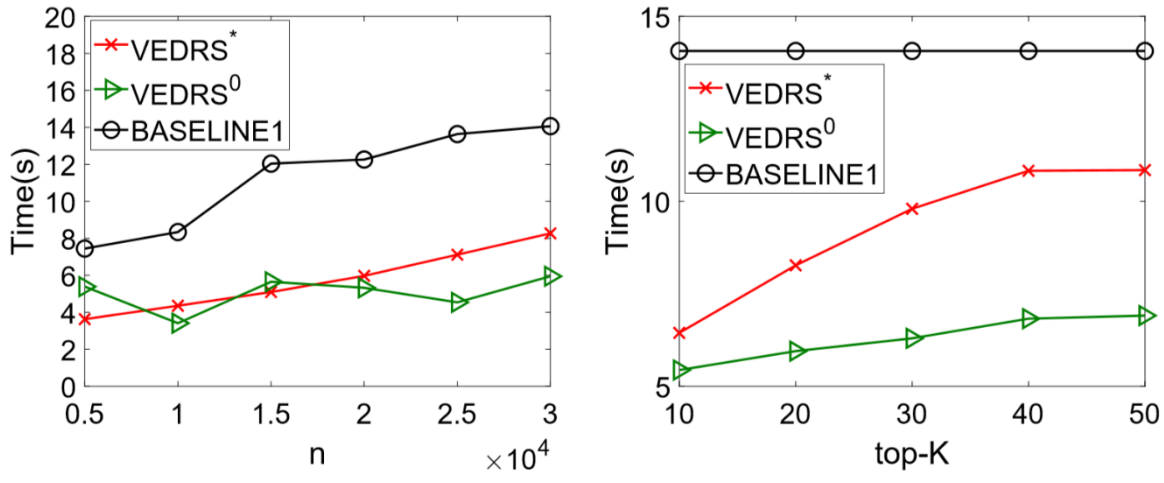
(c) Search (d) GenProof

Fig. 3. Execution time in the search phase (server side)

In Fig. 3-(a), as the number of documents n grows, BASELINE1 needs to decrypt longer strings and return more documents, rendering the execution time to grow linearly; The search time in our VDERS constructions and BASELINE2 is impacted by the number of documents matching a keyword, and thus increases smoothly.

In Fig. 3-(b), the execution time of algorithm GenProof in both VDERS0 and BASELINE is mainly impacted by the number of documents n. However, in our advanced GenProof algorithm, the computation is based on the execution time increases slightly as n increases.

Fig. 3-(c)(d), we know that the execution time of algorithm Search is slightly impacted by parameter K, but the execution time of algorithm GenProof decreases smoothly as K increases.



ISSN: 2278-4632
Vol-10 Issue-6 No. 7 June 2020

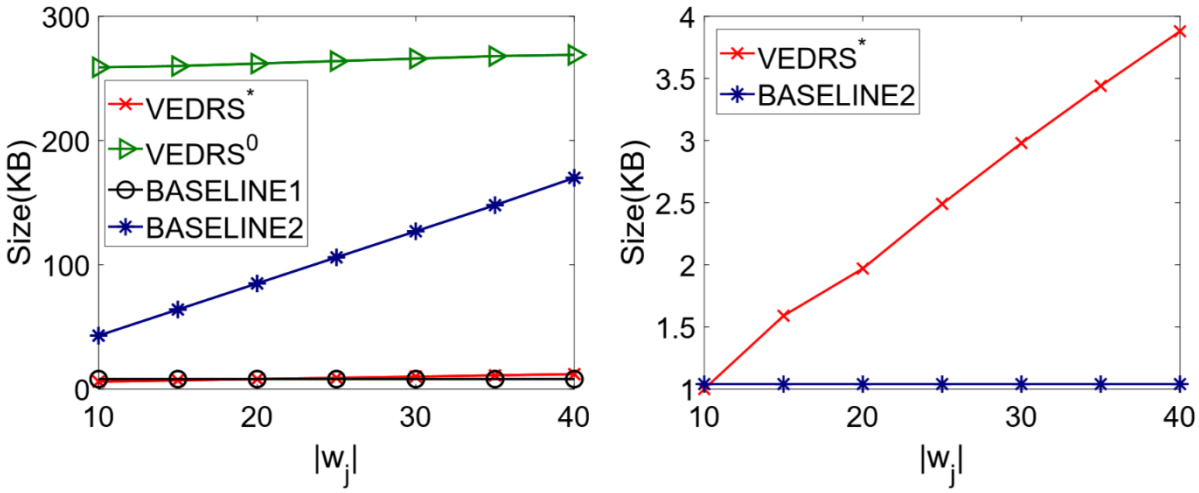
(a) Verify (b) Verify

Fig. 4. Execution time in the search phase (user side)

Fig. 4 shows the execution time at the user side in the search phase. From this figure, we know that our VDERS constructions have better performance compared with BASELINE1. To verify the search results, the number of RSA accumulator calculations in BASELINE1, VDERS^0, and VDERS⋆ is mainly impacted by $|d_W|$+n, K, and $|d_W|$+K, respectively.

Therefore, the execution time of BASELINE1 in Fig. 4-(a) grows linearly as n increases, but it remains unchanged in Fig. 4-(b) as K increases. On the contrary, the execution time of VDERS⋆ in Fig. 4-(a) changes slightly and VDERS0 just fluctuates around a fixed value as n increases, but they both grow linearly in Fig. 4-(b) as K increases.

Fig. 5 shows the communication costs in the update phase. In terms of adding a document, VDERS0 incurs the largest communication cost because it needs to upload m extra random numbers to the CSP. BASELINE1 incurs the minimal cost because it only needs to upload m bits.



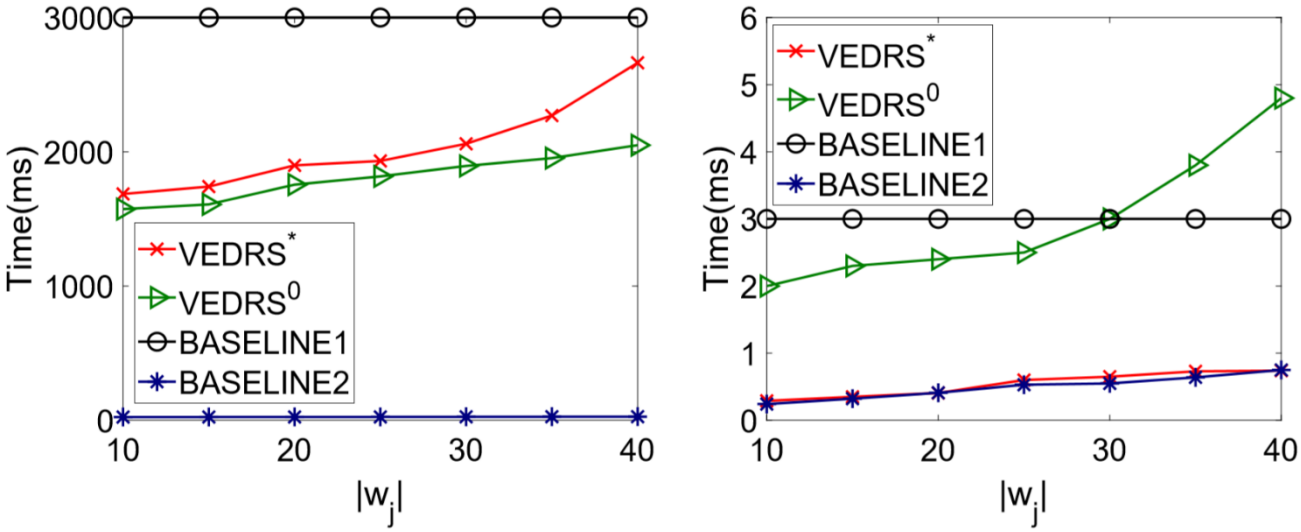
(a) Communication cost (add) (b) Communication cost (delete)

Fig. 5. Communication cost in the update phase

From Fig. 5-(a), we know that the cost of BASELINE2 grows linearly, the costs of our VDERS constructions grow slightly, but the cost of BASELINE1 remains unchanged, as |wj| increases. From Fig. 5(b), we know that the cost of BASELINE2 is independent of |wj|, but the cost of VDERS⋆ badly grows with|wj|.

ISSN: 2278-4632
Vol-10 Issue-6 No. 7 June 2020

Fig. 6 shows the computation time in the update phase.

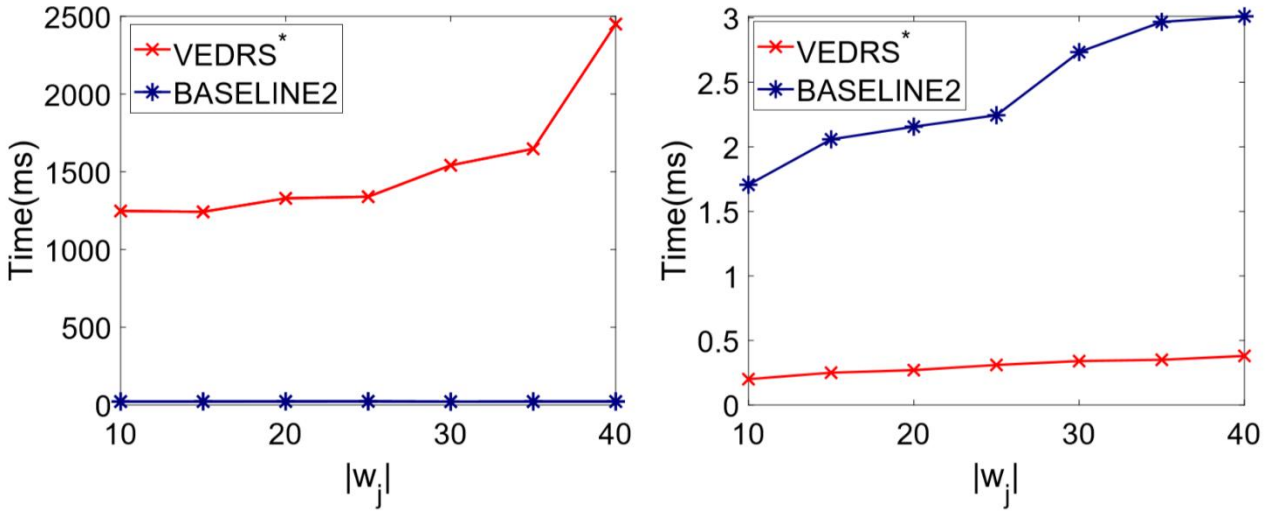


(a) UpdToken (add) (b) Update (add)

(a) UpdToken (delete) (b) Update (delete)

Fig. 6. Execution time in the update phase

In Fig. 6-(a), BASELINE1 needs to compute m times of RSA accumulators, and thus it incurs the maximum execution time. In Fig. 6-(b), VDERS^0 incurs the maximum execution time since it needs to update both the ranked inverted index and the verifiable matrix. In terms of deleting a document, we still only consider VDERS⋆ and BASELINE2. In Fig. 6(c), the time of BASELINE2 is independent of |wj|, but the time of VDERS⋆ grows with |wj|. In Fig. 6-(d), both the time of VDERS⋆ and BASELINE2 grows as |wj| increases. Furthermore, BASELINE2 with a more complicated index structure incurs more execution time than VDERS⋆.



ISSN: 2278-4632
Vol-10 Issue-6 No. 7 June 2020

5. CONCLUSION

In this paper, we design a VDERS scheme to simultaneously support sublinear search time, efficient update and verification, and on-demand information retrieval in cloud computing environment. Experiment results demonstrate that our scheme is effective for verifying the correctness of top-K searches on a dynamic document collection. As part of our future work, we will try to design a forward secure VDERS scheme, where the update phase does not leak keyword information about a newly added document.

REFERENCES

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

[2] Q.Liu,X.Nie,X.Liu,T.Peng,andJ.Wu,"Verifiablerankedsearch 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, Σοφος: 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.Kurosawaa and Y.Ohtaki,"How to update documents verifiably in searchable symmetric encryption," in Proc. of CNS, 2013.

[17] Viswanth, V. S., Ramanujam, R., and Rajyalakshmi, G. "A review of research scope on sustainable and eco-friendly electrical discharge machining (E-EDM)". Materials Today: Proceedings, 5(5), 12525-12533, 2018.

[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] Viswanth, V. S., Ramanujam, R., and Rajyalakshmi, G, "A Novel MCDM Approach for Process Parameters Optimization in Eco-Friendly EDM of AISI 2507 Super Duplex Stainless Steel". Journal of Advanced Research in Dynamical and Control Systems - JARDCS, vol. 10, No. 7, 54-64, 2018.

[20] 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.

[21] 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.

Author's Profile:

D. Saritha Reddy has received her M.Tech degree in Computer Science from Acharya Nagarjuna University, Guntur in 2010 and pursuing Ph.D with area Computer Networks. At present she is working as Assistant Professor in Narayana Engineering College, Gudur, Andhra Pradesh, India.

M. Lokesh has Received his B.Sc Electronics from Swarnandra Bharathi Degree College, Gudur affiliated to Vikrama Simhapuri University, Nellore in 2017 and

ISSN: 2278-4632
Vol-10 Issue-6 No. 7 June 2020

pursuing PG Degree in Master of Computer Applications (M.C.A) from Narayana Engineering College, Gudur affiliated to JNTU Anantapur, Andhra Pradesh, India.

