AN EFFICIENT CIPHERTEXT INDEX RETRIEVAL SCHEME BASED ON EDGE COMPUTING FRAMEWORK FINAL DOCUMENT

1N.RAJITHA, 2A.D.SIVARAMA KUMAR, 3M.N.MALLIKARJUNA REDDY,

1STUDENT, 2ASSISTANT PROFESSOR, 3ASSOCIATE PROFESSOR

DEPARTMENT OF CSE

SVR ENGINEERING COLLEGE AYYALURU,NANDYAL, ANDHRA PRADESH 518503

ABSTRACT:

With the rapid development of mobile applications, more and more traffic is generated at the network's edge and forwarded between many users. The explosive growth of network traffic has imposed massive pressure on traditional network architectures. At the same time, users have increasing data security requirements because of frequent data breaches. Mobile edge storage is an emerging computing framework that ensures users enjoy a high quality of experience when they access cloud services and is gradually becoming the key technology to solve the above problems. In this paper, by exploiting searchable encryption and cooperative edge computing, we proposed an efficient ciphertext index retrieval scheme to tackle three issues simultaneously in a secure and efficient data search service scenario: (1) reducing data transportation latency to improve mobile user's quality of experience; (2) mitigating data traffic pressure on the backbone network; (3) guaranteeing the security of the data when users search data in the edge network. Simulation results show that our scheme can save about 80% of backbone network traffic than the traditional cloud computing scheme. It can also reduce network latency by approximately 30% for users.

I. INTRODUCTION

1 About the Project

With the rapid development of mobile applications, mobile communications have generated more and more traffic between network edges and users. The explosive growth of network traffic has put tremendous pressure on traditional network architecture. Mobile edge storage is an emerging computing framework when users enjoy a high-quality experience when

accessing cloud services. It has gradually become a key technology to solve the above problem. Forecasts predict that the world's mobile data traffic will surpass 90% of mobile data traffic and reach 77.5 monthly ex a bytes by 2022 [1]. Such explosive traffic has exerted a heavy burden on the current network architecture [2]. With the frequent interaction of these massive data, information leakage and data privacy have gradually become the focus of attention in mobile edge storage system.

As a new encryption technology, searchable encryption is favored by governments and enterprises, which enables users to search over encrypted data without exposing the contents of messages or the searched keyword to cloud storage operators. This has a huge appeal for governments and companies with a need for secrecy. Through many researchers' efforts in the field of encryption, many efficient searchable encryption schemes have been formed [3]–[23].

An important research direction of searchable encryption is symmetric searchable encryption (SSE), which has a low computational cost, reduced algorithm, fast speed, and is closer to the practical application scene. A classic SSE system is as follows [24]:

1) Users first extract keywords from local files and construct the index, then encrypt the index and files with the private key, and then upload them to the cloud server.

2) Users with query permissions use the private key to generate a trapdoor for the keyword that needs to be queried and then sent to the server. The trapdoor cannot reveal any information about the keyword. The simplest trapdoor is the keyword encrypted with the key.

3) The server executes the retrieval algorithm after receiving the trapdoor. The retrieval algorithm finds the encrypted file name corresponding to the trapdoor's keyword from the inverted index structure. Finally, the encrypted file corresponding to the user's file name is returned. The server can only know whether the encrypted file has a keyword contained in the trapdoor.

4) Users use the key to decrypt the encrypted file returned by the server to obtain query results.

In the above process, the user can obtain the cipher text's query authority by sharing the key and other methods. The user's key keep secret from the third party.

The server neither knows the user's key nor the file content. So the user's encrypted file content is safe. Figure 1 shows the specific SSE process. However, most of the traditional SSE is based on cloud storage mode. Obviously, the cloud server undertakes a large number of search operations, and the file transmission has a large transmission overhead. In the mobile edge network, the base station is close to the user's site, which has a certain storage capacity and computing power. In this paper, we use the resources of the edge base station to cache encrypted file index to provide search efficiency.

Symmetric searchable encryption usually combines with a variety of encryption techniques. For example, the combination of symmetric searchable encryption and attribute based encryption technology [18], [25] can make the combined scheme have the function of querying by fine-grained attributes. The combination of symmetric searchable encryption and proxy re-encryption [26] can enable the combined scheme to realize a multi-user search function. It should be pointed out that multi-user and multi-keyword often increase the complexity of the system. How to prevent the cost of encryption algorithm increase with the increase of users, which also has some work [27].

Current work on searchable encryption has focused on the single-server model. All data store in a single server rather than multiple servers. Users tend to choose multiple cloud service providers at the same time to ensure data security, while cloud service providers tend to provide services to more users to make money. Therefore, the multi-server multi-user model is one of the future development trends of searchable encryption [17]. The multi-server multi-user model's difficulties are as follows: keyword retrieval involves multiple un trusted objects; [28] exists key leakage problem under multi-user sharing; the multi-user single server model has the query result ordering problem the safety of PEKS.

With the rapid increase in the number of users with searchable encryption, the traditional server-centric service model will not cope with such a large number of requests. Searchable encryption in the encryption, update, retrieval phase will have more energy consumption. Computing migration and content caching in mobile edge networks must diffuse the pressure on the central servers. At present, the mainstream searchable encryption scheme

bases on symmetric searchable encryption (SSE). By constructing the keyword of custom cipher text into an inverted index [19], the user can find the trapdoor and the corresponding cipher text in the inverted index according to the cipher text of the user's keywords when querying. An index can be split into multiple units or merged into a single global index [29]. Therefore, with SSE, index can be naturally divided and cached in the edge network.

As we know, the user's request conforms to Zipf Law [30] Users of the same network will often access the same index item. The same user will repeatedly request the same index item. Therefore, there are numerous application scenarios [25] in the cache edge network. Suppose the base station in the edge network near the user happens to have the index item needed by the user. In that case, the user does not need to get it from the remote central server, which significantly saves the traffic [31] in the edge network. For the central server, reducing the number of direct requests from users, the server's pressure and cost can be significantly reduced. The service provider saves the traffic cost paid to the network operator by reducing the backbone network traffic. Chen et al. [32] discussed in detail a partial migration decision problem in a multi-mobile device scenario. The author abstracts it into a solvable linear programming problem. The best solution obtained by exhaustive can reduce the amount of computation by 40% compared with no strategy. Searchable encrypted index entries are like static data cached by Content Delivery Network (CDN) in [33], and there can be many caching algorithms that ultimately reduce carrier traffic.

One approach is to use online technology that selects a specific cache item [25], [34] based on recent requests. It has the advantage of being able to change itself in more real-time to respond to user requests. The impact of caching on request delays can be felt more intuitively by users. Online technology is necessary for systems with higher real time requirements. One problem with this technology is that base stations take up a lot of computing power and cannot integrate with other base stations. An alternative approach is to use the offline technique to preload possible index entries [35] on a base station close to the user. The base station can choose to cache index entries when the network is idle. The optimal cache algorithm is an NPC problem. If the optimal solution is required, it will consume too much computing resources. In order to improve the effectiveness of

caching, there is much work on caching algorithms in edge networks. The number of indexes that the base station can cache is much smaller than the number of global indexes that the central server has. The number of indexes that the base station can cache and its strategy to cache data will significantly af ect the edge network's performance [36]. Some of the work has focused on creating a collaborative cache [34], [37] which collaborates between edge devices to form a connection network [35], [38] with minimal latency and energy consumption.

With the rapid development of mobile applications, more and more traffic is generated at the network's edge and forwarded between many users. The explosive growth of network traffic has imposed massive pressure on traditional network architectures. At the same time, users have increasing data security requirements because of frequent data breaches. Mobile edge storage is an emerging computing framework that ensures users enjoy a high quality of experience when they access cloud services and is gradually becoming the key technology to solve the above problems. In this paper, by exploiting searchable encryption and cooperative edge computing, we proposed an efficient ciphertext index retrieval scheme to tackle three issues simultaneously in a secure and efficient data search service scenario:

(1) reducing data transportation latency to improve mobile user's quality of experience;

(2) mitigating data traffic pressure on the backbone network;

(3) guaranteeing the security of the data when users search data in the edge network. Simulation results show that our scheme can save about 80% of backbone network traffic than the traditional cloud computing scheme. It can also reduce network latency by approximately 30% for users.

II. EXISTING SYSTEM

- X. Liu et al. [9] and Jie Cui et al. [41] proposed an encrypted search scheme that allowed multi-user to search in cloud storage. Users could generate private keys for encryption, and other users could query the encrypted content via the public key. As the number of users increases, it tended to increase the load on the server. It was very unfriendly to a service provider of searchable encryption. S. Li et al. [22] proposed a potentially secure multi-user multi-

keyword searchable encryption scheme. This approach did not increase the data owner's encryption workload as the number of data users increases. To improve the accuracy of multikeyword fuzzy search, Zhong et al. [42] develop an index tree and top-k search algorithm.

- To reduce the load on the cloud server, M. B. Mollah et al. [43] proposed a secure data search and sharing scheme. The scheme implemented the weak trust hypothesis on edge devices. It also supported delegating computation-intensive encryption and decryption to edge devices, which reduced the overhead of keyword search latches. Zhong H et al. [44] proposed a two-stage index-based central-keyword ranked search scheme. The scheme can reduce the computation cost in the query process. For multiple query keywords, the scheme's search results are more accurate.
- S. Wang et al. [16] used fog computing to distribute encrypted essential data to multiple fog nodes of an edge network. To support the distribution of encrypted data, Curtmola et al. [24] studied the trusted data outsourcing mechanism in encrypted search. This mechanism could send encrypted data security scores to untrusted objects. Cui et al. [45] use online/offline ABE technology and outsourcing technology to reduce the online calculation cost and the local calculation cost of mobile users. Some authors tried to apply blockchain to encrypted search [10], which improved the efficiency of search.

DISADVANTAGES

- There is less security on outsourced data due to lack of trapdoor generation on outsourced data.
- There is no Data Integrity Proof on outsourced data.

III. PROPOSED SYSTEMS

In this project, we propose a cache placement algorithm, which combined a mobile edge network model and an edge network with game theory. This algorithm solves the searchable encrypted index caching problem in the edge network. In our algorithm, we will build a mobile edge network model based on game theory algorithms [36]. Each node of the network is a base station, which can store index entries.

Each node is responsible for maximizing its interests. Our primary researches are as follows:

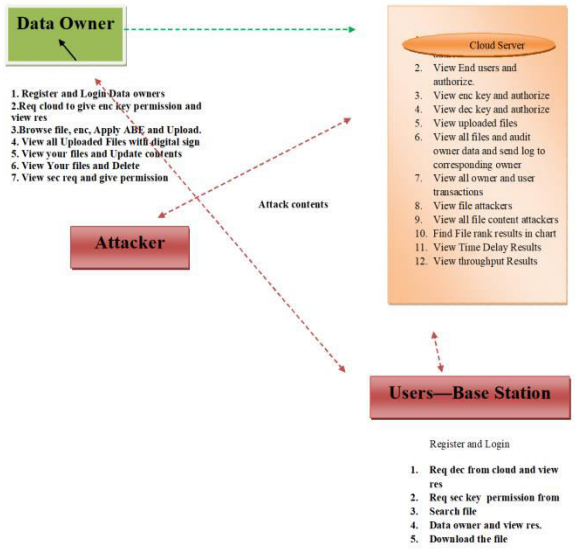
1) We define an edge caching network model. The network contains many base station nodes. Base stations can provide services for mobile users. Every base station in the edge network can keep a limited number of indexes. Base stations can get indexes from other base stations that they do not have. If the requested index does not exist in any base station, the base station will obtain it from the central server.

2) We define a game theory model to find spontaneous cooperation phenomena in non-cooperative games. In our game, every base station is a player. Each player can decide what index to store and modify its strategy based on the data currently held by other players. If a player asks other players to provide data, they need to pay. In the game, we find the Nash equilibrium by maximizing the utility of each player. After further analysis, we find a pure Nash equilibrium. At the Nash equilibrium point, we can obtain a cache placement scheme for the locally optimal solution. By analyzing each player's cache strategy, we can find that there is a spontaneous collaboration between the caches. Finally, we design a distributed algorithm that can achieve balance.

ADVANTAGES

- The proposed system implements the central server stores a large amount of ciphertext index information. Users can get whatever content they want from the central server.
- The system is more effective due to presence of the base station which collects the trapdoor of the keyword in the user's request when querying the encrypted file and the index of the file in the server's response

IV. SYSTEM ARCHITECTURE



V. MODULES

Data Owner

In this module, the data owner uploads their data in the cloud server. For the security purpose the data owner encrypts the file and the index name and then store in the cloud. The data encryptor can have capable deleting of a specific file. And also he can view the transactions based on the files he uploaded to cloud and will do the following operations like Register and Login Data owners,Req cloud to give enc key permission and view res,Browse file, enc, Apply ABE and Upload, View all Uploaded Files with digital sign, View your files and Update contents, View Your files and Delete , View sec req and give permission.

.**Users—Base Station**

In this module, user logs in by using his/her user name and password. After Login user requests search control to cloud and will Search for files based on the index keyword with the Score of the searched file and downloads the file. User can view the search of the files and also do some operations like Req dec from cloud and view res,Req sec key permission from, Search file ,Data owner and view res,Download the file.

Cloud Server

The cloud server manages a cloud to provide data storage service. Data owners encrypt their data files and store them in the cloud for sharing with Remote User. To access the shared data files, data consumers download encrypted data files of their interest from the cloud and then decrypt them.

The cloud server authorizes the data owner and the data user and provides the search requests sent from the users. Also in this module it shows personalized search model and the interest search model. Can view all the file attackers and doing following operations View data owners and authorize, View End users and authorize, View enc key and authorize, View dec key and authorize, View uploaded files, View all files and audit owner data and send log to corresponding owner, View all owner and user transactions, View file attackers, View all file content attackers, Find File rank results in chart, View Time Delay Results, View throughput Results.

VI. CONCLUSION

In this project, we propose a game-theoretic caching strategy for indexing in mobile edge networks. Through our cache placement algorithm, each edge node could select its caching strategy in advance, effectively alleviating the traffic pressure on the backbone network. Compared with the global optimal solution algorithm, the game theory algorithm's hit rate has increased by

about 10%, and the average node revenue has increased by about 20%. Compared with traditional cloud computing, our solution saves about 80% of backbone network Traffic and 30% of network latency. In further work, we will study how this paper's caching strategy ensures security under heterogeneous infrastructure [55]. In addition, we will study the sorting optimization problem in distributed caches in the future.

REFERENCES

[1] Forecast, Global Mobile Data Traffic. "Cisco visual networking index:

global mobile data traffic forecast update, 2017–2022." Update 2017 (2019): 2022.

[2] E. Bastug, M. Bennis and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks, " in IEEE Communications Magazine, vol. 52, no. 8, pp. 82-89, Aug. 2014, doi: 10.1109/MCOM.2014.6871674.

[3] A. Awad, A. Matthews, Y. Qiao and B. Lee, "Chaotic Searchable Encryption for Mobile Cloud Storage, " in IEEE Transactions on Cloud Computing, vol. 6, no. 2, pp. 440-452, 1 April-June 2018, doi: 10.1109/TCC.2015.2511747.

[4] P. Xu, S. He, W. Wang, W. Susilo and H. Jin, "Lightweight Searchable Public-Key Encryption for Cloud-Assisted Wireless Sensor Networks, " in IEEE Transactions on Industrial Informatics, vol. 14, no. 8, pp. 3712- 3723, Aug. 2018, doi: 10.1109/TII.2017.2784395.

[5] Y. Yao, Z. Zhai, J. Liu and Z. Li, "Lattice-Based Key-Aggregate (Searchable) Encryption in Cloud Storage, " in IEEE Access, vol. 7, pp. 164544- 164555, 2019, doi: 10.1109/ACCESS.2019.2952163.

[6] X. Liu, G. Yang, W. Susilo, J. Tonien, R. Chen and L. Xixiang, "Message-Locked Searchable Encryption: A New Versatile Tool for Secure Cloud Storage, " in IEEE Transactions on Services Computing, doi: 10.1109/TSC.2020.3006532.

[7] K. Li, W. Zhang, C. Yang and N. Yu, "Security Analysis on One-to- Many Order Preserving Encryption-Based Cloud Data Search, " in IEEE Transactions on Information Forensics and Security, vol. 10, no. 9, pp. 1918-1926, Sept. 2015, doi: 10.1109/TIFS.2015.2435697.

[8] L. Zhang, J. Su and Y. Mu, "Outsourcing Attributed-Based Ranked Searchable Encryption With Revocation for Cloud Storage, " in IEEE Access, vol. 8, pp.

104344-104356, 2020, doi: 10.1109/ACCESS. 2020.3000049.

[9] X. Liu, G. Yang, Y. Mu and R. Deng, "Multi-user Verifiable Searchable Symmetric Encryption for Cloud Storage, " in IEEE Transactions on Dependable and Secure Computing, doi: 10.1109/TDSC.2018.2876831.

[10] J. Niu, X. Li, J. Gao and Y. Han, "Blockchain-Based Anti-Key-Leakage Key Aggregation Searchable Encryption for IoT, " in IEEE Internet of Things Journal, vol. 7, no. 2, pp. 1502-1518, Feb. 2020, doi: 10.1109/JIOT.2019.2956322.

[11] H. Wang, J. Ning, X. Huang, G. Wei, G. S. Poh and X. Liu, "Secure Fine-grained Encrypted Keyword Search for e-Healthcare Cloud, in IEEE Transactions on Dependable and Secure Computing, doi: 10.1109/TDSC.2019.2916569.

[12] Y. Yang, "Towards Multi-user Private Keyword Search for Cloud Computing,

" 2011 IEEE 4th International Conference on Cloud Computing, Washington, DC, 2011, pp. 758-759, doi: 10.1109/CLOUD.2011.76.

[13] A. Xiong, Q. Gan, X. He and Q. Zhao, "A searchable encryption of CP-ABE scheme in cloud storage, " 2013 10th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, 2013, pp. 345-349, doi: 10.1109/ICCWAMTIP. 2013.6716664.

[14] J. Huang and I. Liao, "A searchable encryption scheme for outsourcing cloud storage, " 2012 IEEE International Conference on Communication, Networks and Satellite (ComNetSat), Bali, 2012, pp. 142-146, doi:

10.1109/ComNetSat.2012.6380794.

[15] Y. Miao, J. Ma, X. Liu, J. Weng, H. Li and H. Li, "Lightweight Fine- Grained Search Over Encrypted Data in Fog Computing, " in IEEE Transactions on Services Computing, vol. 12, no. 5, pp. 772-785, 1 Sept.- Oct. 2019, doi: 10.1109/TSC.2018.2823309.