# A Protected and Fine-Grained Query Results Verification Scheme for Search over Encrypted Cloud Data

**K.VENKATESWARI & MR. I PHANI KUMAR**
1PG Scholar, Dept of CSE, VelagaNageswaraRao College Of Engineering,
Ponnur(Post),Ponnur(Md)Guntur(D.T), Andhra Pradesh

2Assoc Professor, Dept of CSE, VelagaNageswaraRao College Of Engineering,
Ponnur(Post),Ponnur(Md)Guntur(D.T)A. Andhra Pradesh

## Abstract:

In this project, we design a secure, easily integrated, and fine-grained query results verification mechanism, by which, given an encrypted query results set, the query user not only can verify the correctness of each data file in the set but also can further check how many or which qualified data files are not returned if the set is incomplete before decryption. The verification scheme is loose-coupling to concrete secure search techniques and can be very easily integrated into any secure query scheme. We achieve the goal by constructing secure verification object for encrypted cloud data. Furthermore, a short signature technique with extremely small storage cost is proposed to guarantee the authenticity of verification object and a verification object request technique is presented to allow the query user to securely obtain the desired verification object. Performance evaluation shows that the proposed schemes are practical and efficient.

## I. Introduction

Cloud computing is a model for enabling ubiqui-tous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with mini-mal management effort or service provider interaction [1]. Driven by the abundant benefits brought by the cloud computing such as cost saving, quick deployment, flexible resource configuration, etc., more and more en-terprises and individual users are taking into account migrating their private data and native applications to the cloud server. A matter of public concern is how to guarantee the security of data that is outsourced to a remote cloud server and breaks away from the direct control of data owners [2]. Encryption on private data before outsourcing is an effective measure to protect data confidentiality [3]. However, encrypted data make

Effective data retrieval a very challenging task.

To address the challenge (i.e., search on encrypted allows users to search over encrypted data through encrypted query keywords in [4]. Later, many searchable encryption schemes were proposed based on symmetric key and public-key setting to strengthen security and improve query efficiency [5], [6], [7], [8], [9], [10], [11], [12]. Recently, with the growing popularity of cloud computing, how to securely and efficiently search over encrypted cloud data becomes a research focus. Some approaches have been proposed based on traditional searchable encryption schemes in [13], [14], [15], which aim to protect data security and query privacies with better query efficient for cloud computing. However, all of these schemes are based on an ideal assumption that the cloud server is an ”honest-but-curious” entity and keeps robust and secure software/hardware environments. As a result, correct and complete query results always be unexceptionally returned from the cloud server when a query ends every time. However, in practical applications, the cloud server may return erroneous or incomplete query results once he behaves dishonestly for illegal profits such as saving computation and communication cost or due to possible software/hardware failure of the server .

## II. Related Work

### Secure Search in Cloud Computing

Essentially, the secure search is thus a technique that allows an authorized data user to search over the data owner’s encrypted data by submitting encrypted query keywords in a privacy-preserving manner and is an effective extension of traditional searchable encryption to adapt for the cloud computing environment. Moti-vated by the effective information retrieve on encrypted outsourced cloud data, Wang et al. first proposed a keyword-based secure search scheme [13] and later the secure keyword search issues in cloud computing have been adequately researched which aim to continually improve search efficiency, reduce communication and computation cost, and enrich the category of search function with bet-ter security and privacy protection. A common basic assumption of all these schemes is that the cloud is considered to be an ”honest-but-curious” entity as well as always keeps robust and secure software/hardware environments. As a result, under the ideal assumption, the correct and complete query results always be unex-ceptionally returned from the cloud server when a query ends every time.

### Verifiable Secure Search in Cloud Computing

In practical applications, the cloud server may return erroneous or false search results once he behaves dis-honestly for illegal profits or due to possible soft- ware/hardware failure of the cloud server. Because of the possible data corruption under a dishonest setting, serval research works have been proposed to allow the data user to enforce query results verification in the secure search fields for cloud computing. In , Wang et al. applied hash chain technique to implement the com-pleteness verification of query results by embedding the encrypted verification information into their proposed secure searchable index. In  Sun et al. used encrypted index tree structure to implement secure query result-s verification functionality. In this scheme, when the query ends, the cloud server returns query results along with a minimum encrypted index tree, then the data user searches this minimum index tree using the same search algorithm as the cloud server did to finish result verification. Zheng et al. constructed a verifiable secure query scheme over encrypted cloud data based on attribute-based encryption technique (ABE) in the public-key setting. Sun et al. referred to the Merkle hash tree and applied Pairing operations to implement the correctness and completeness verification of query results for keyword search over large dynamic encrypted cloud data. However, these secure verification schemes cannot achieve our proposed fine-grained verification goals. Furthermore, these verification mechanisms are generally tightly coupled to corresponding secure query schemes and have not universality.

## III. Implementation

### System Framework:

In this framework, we design a secure, easily integrated, and fine-grained query results verification mechanism, by which, given an encrypted query results set, the query user not only can verify the correctness of each data file in the set but also can further check how many or which qualified data files are not returned if the set is incomplete before decryption. The verification scheme is loose-coupling to concrete secure search techniques and can be very easily integrated into any secure query scheme. We achieve the goal by constructing secure verification object for encrypted cloud data. Furthermore, a short signature technique with extremely small storage cost is proposed to guarantee the authenticity of verification object and a verification object request technique is presented to allow the query user to securely obtain the desired verification object. Performance evaluation shows that the proposed schemes are practical and efficient. Here we implement some modules they are Data Owner, Data User and Cloud Server.

### Data Owner:

In Data Owner module, Initially Data Owner must have to register their detail. After successful registration data owner can login and upload files into cloud server with encrypted keywords and hashing algorithms. He/she can view the files that are uploaded in cloud. Data Owner can approve or reject the file request sent by data users. After request approval data owner will send the trapdoor key and verification object through mail.

### Data User:

In Data User module, Initially Data Users must have to register their detail and after login he/she has to verify their login through secret key. Data Users can search all the files upload by data owners. He/she can send request to the files and then request will send to the data owners. If data owner approve the request then he/she will receive trapdoor, verification object and decryption key in registered mail

### Cloud Server:

In Cloud Server module, Cloud Provider can view all files details. Cloud can edit the files and update and also cloud server can view the download history

## IV.The Verification Object Construction

To maximize reduce storage and communication cost and achieve privacy guarantee of the verification objects, in this paper, we will utilize Counting Bloom Filters and the pseudo-random function $prf_k$ to construct our verification objects, on which the authorized data user can efficiently perform query results verification. Next, we elaborate on the construction process of verification objects as follows. Given a ciphertext set $C_w$ of $F_w$, the data owner first generates a Counting Bloom Filter V $O_w$ with m counters, in which each counter is set to be 0 initially. Then, for each encrypted data file c 2 $C_w$, he uses the pseudo-random function $prf_k$ under the key k to calculate the secret value $prf_k(c)$. Further, he continues to uses l hash functions $h_1; :::; h_l$ of the hash function family H to hash the $prf_k(c)$ to get $h_1(prf_k(c)$ 2 [0; m 1]; :::; $h_l(prf_k(c))$ 2 [0; m 1]. Lastly, the data owner inserts these hash values into the Counting Bloom Filter V $O_w$ by performing operations that the corresponding counter V $O_w[h_i(prf(c))]$; 1 i l is increased by 1. Our basic idea is to let the V $O_w$ represent the verification object of $C_w$.

**Algorithm 1** Constructing Verification Objects

```
Input:
    The ciphertext files collection C = f$C_w$g$_{w2W}$
Output:
    The verification objects collection fV O$_w$g$_{w2W}$
Generate an empty set VO=fg;
for each $C_w$ 2 C do
    Generate a Counting Bloom Filter V $O_w$ with n counters;
    for each c 2 $C_w$ do
        Calculate $v_c = prf_k(c)$;
        Calculate $h_1(v_c); :::; h_l(v_c)$ using hash function family H;
        V $O_w[h_1(v_c)]$; :::; V $O_w[h_l(v_c)]$ are increased by 1 in V $O_w$;
    end for
    Generate l (j$C_w$j$_{max}$ j$C_w$j) random strings $R_1; R_2$; :::
    Calculate $P(R_1); P(R_2)$; :::
    V $O_w[P(R_1)]$; V $O_w[P(R_2)]$; ::: are increased by 1 in
    V $O_w$;
    Add the V $O_w$ into VO
end for
return VO = fV O$_w$g$_{w2W}$
```

## V. Conclusion

In this paper, we propose a secure, easily integrated, and fine-grained query results verification scheme for secure search over encrypted cloud data. Different from previous works, our scheme can verify the correctness of each encrypted query result or further accurately find out how many or which qualified data files are returned by the dishonest cloud server. A short signature technique is designed to guarantee the authenticity of verification object itself. Moreover, we design a secure verification object request technique, by which the cloud server knows nothing about which verification object is requested by the data user and actually returned by the cloud server. Performance and accuracy experiments demonstrate the validity and efficiency of our proposed scheme.

## References

[1] P. Mell and T. Grance, "The nist definition of cloud computing," http://dx.doi.org/10.602/NIST.SP.800-145.

[2] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69–73, 2016.

[3] S. Kamara and K. Lauter, "Cryptographic cloud storage," in Springer RLCPS, January 2016.

[4] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in IEEE Symposiumon Security and Privacy, vol. 8, 2016, pp. 44–55.

[5] E.-J.Goh, "Secure indexes," IACR ePrint Cryptography Archive, http://eprint.iacr.org//2016, Tech. Rep.,.

[6] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public-key encryption with keyword search," in EUROCRYPR, 2015, pp. 506–522.

[7] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved deinitions and efficient constructions," in ACM CCS, vol. 19, 2016, pp. 79–88.

[8] M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and efficiently searchable encryption," in Springer CRYPTO, 2015.

[9] K. Kurosawa and Y. Ohtaki, "Uc-secure searchable symmetric encryption," Lecture Notes in Computer Science, vol. 7397, pp. 258–274, 2015.

[10] P. Xu, H. Jin, Q. Wu, and W. Wang, "Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack," IEEE Transactions on Computers, vol. 62, no. 11, pp. 2266–2277, 2015.

[11] S. Kamara and C. Papamanthou, "Parallel and dynamic searchable symmetric encryption," in Financial Cryptography and Data Security. Springer Berlin Heidelberg, pp. 258–274.

[12] M. Naveed, M. Prabhakaran, and C. A. Gunter, "Dynamic searchable encryption via blind storage," in IEEE S&P, May 2014, pp. 639–654.

[13] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in IEEE ICDCS, 2014, pp. 253–262.

## Author Details

**K.VENKATESWARI**

pursing M. Tech in Computer Science and Engineering from VelagaNageswaraRao College Of Engineering in 2018, respectively.

ketinetivenkateswari7@gmail.com
**8790504565**

**I.PHANI KUMAR** received M.Tech in Computer Science He is currently working as Assoc. Professor And HOD, Dept of C.S.E, VelagaNageswaraRao College Of EngineeringCollege,Ponnur(Post),Ponnur(Md)Guntur(D.T), Andhra Pradesh, A.P, and India.phankumari@gmail.com

9110323715