www.ijemr.net ISSN (ONLINE): 2250-0758, ISSN (PRINT): 2394-6962

Vandana Publications
IJEMR

Volume-7, Issue-2, March-April 2017
International Journal of Engineering and Management Research
Page Number: 408-413

Secured Novel Additive Protocol to Rank Multi Keyword Search in Multiple Data Owners in Cloud Computing

Alka Kulkarni[1], Rohini Bhosale[2]
[1]Computer Engineering Department, KJEI's Trinity College of Engineering and Research, Pune, INDIA
[2]Computer Engineering Department, KJEI's Trinity College of Engineering and Research, Pune, INDIA

ABSTRACT

With the Emergence of cloud computing, it has become increasingly admire for data owners to outsource their data to public cloud servers while allowing data users to retrieve this data. For privacy concerns, authenticated secure searches over encrypted cloud data have motivated several research works under the single owner model. Although, most of the cloud servers in real time practice do not just serve one data owner instead, they support multiple data owners to share the application and benefits brought by cloud computing. This paper presents different schemes to deal with Privacy preserving Ranked Multi-keyword Search in a Multi-owner model (PRMSM). To permit authorization to cloud servers to perform secure search without knowing the actual data of keywords and trapdoors, unauthorized user we systematically construct a novel secure search protocol. To provide the count of generate number of rank to search results and preserve the privacy of relevance scores between keywords and files, we propose a novel Additive Order and Privacy Preserving Function family. To intercept the unauthorized attackers from eavesdropping secret keys and pretending to be legal data users submitting searches, we propose a novel dynamic secret key generation protocol and a new data user authentication protocol. Furthermore, PRMSM supports efficient data user revocation. The real world extensive experiments on datasets confirm the efficacy and efficiency of PRMSM.

***Keywords*—** Privacy preserving Ranked Multi-keyword Search in a Multi-owner model (PRMSM), Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS), Searchable symmetric encryption (SSE)

I. INTRODUCTION

1.1 Overview of cloud computing

Cloud computing is the type of Internet based system to use computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing provide various remote services with a user's data system software and system hardware form computation. Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services allow various accesses to advanced software applications and high-end networks of server computers as shown in figure 1.

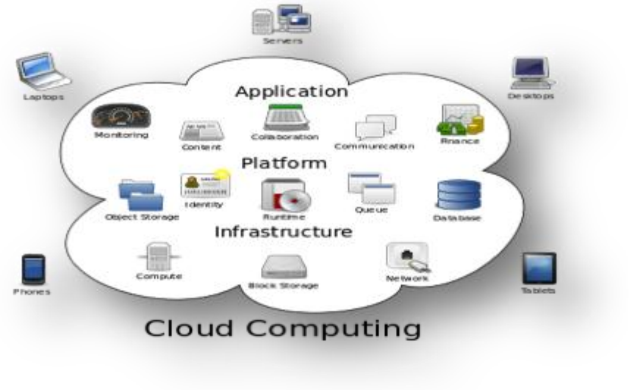

Figure 1: Structure of cloud computing

1.2 Characteristics and working of cloud computing

Cloud computing is used to generate the supercomputing through engineering and scientific calculation to increase the performance of the system. These are used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer game. The National Institute of Standards and Terminology

(NIST) designed the fallowing Features of cloud computing given below.

On demand self Service, Ubiquitous network access that allows various networks to promote use of various client platforms. Cloud provides the Location transparency to computing resources that are pooled together to serve the multi-tenant model. To improve the system performance high-speed elasticity is provided to various networks. And also provide Uniform and measured service is provided each cloud users. Computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical devices.

1.3 Services models:

Cloud Computing designed with various service models which are classified into three different service models, which are namely Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS).

An end user layer that encapsulates the end user perspective on cloud services completes these three-service model or layer. The model is shown in figure 2 below. The cloud user accesses services on the infrastructure layer, for example, they can run their own applications on the resources of a cloud infrastructure and remain responsible for the support, maintenance, and security of these applications their self. If she accesses a service on the application layer, these tasks are normally taken care of by the cloud service provider.

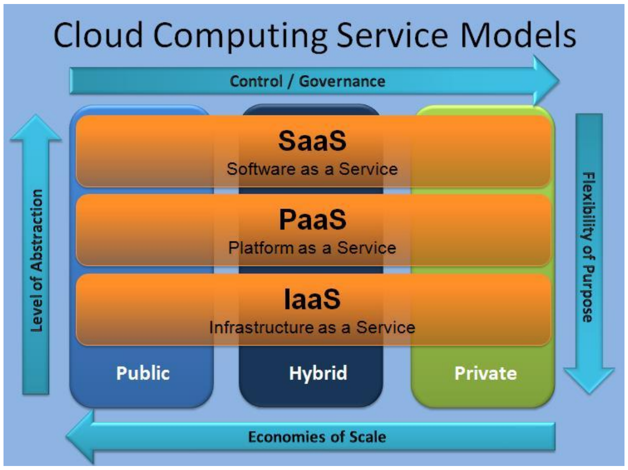

Figure 2: Structure of service models

1.4 Benefits and advantages of cloud computing:

The benefits and advantages of the cloud computing to Achieve economies of scale, Reduce the investment on the technology infrastructure, wide spread the workforce on cheap, improve the performance by doing the working less time. Mainly reduce the capital cost and improve the accessibility and monitor the all projects effectively. Reduce the price, pay only for the resources used. Improve the scalability flexibility and reliability of the services provided by the cloud.

II. LITERATURE SURVEY

2.1 A view of cloud computing

AUTHORS: M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I.Stoica, and M.Zaharia.

Cloud computing has property to design the various real time software application to customize usability of cloud. In real time making software even more attractive as a service and shaping the way IT hardware is designed and purchased. The software developers design various unconventional plans for new Internet services no longer require the large capital outlays in hardware to deploy their service or the human expense to operate it.

The popularity of the cloud computing does not meet their predictions, thus wasting costly resources or under provisioning for one that becomes wildly popular, thus missing potential customers and revenue. In real time most of the IT companies runs the Job in batches to get results as faster and as quickly as their programs can scale, since using 1,000 servers for one hour costs no more than using one server for 1,000 hours. This rapid flexibility and elasticity of resources, without paying a premium for large scale, is unprecedented in the history of IT.

2.2 Privacy preserving public auditing for secure cloud storage.

AUTHORS: C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou

Cloud network allow the cloud user store data remotely and enjoy the on-demand high-quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in cloud computing a formidable task, especially for users with constrained computing resources. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public auditability for cloud storage is of critical importance so that users can resort to a third-party auditor (TPA) to check the integrity of outsourced data and be worry free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities toward user data privacy, and introduce no additional online burden to user.

In this paper, we propose a secure cloud storage system supporting privacy-preserving public auditing. We further extend our result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient. Our preliminary experiment conducted on Amazon EC2 instance further demonstrates the fast performance of the design.

2.3 Practical techniques for searches on encrypted data

AUTHORS: D.Song, D.Wagner, and A.Perrig

It is desirable to store data on data storage servers such as mail servers and file servers in encrypted form to reduce security and privacy risks. But this usually implies that one has to sacrifice functionality for security. For example, if a client wishes to retrieve only documents containing certain words, it was not previously known how to let the data storage server perform the search and answer the query, without loss of data confidentiality. We describe our cryptographic schemes for the problem of searching on encrypted data and provide proofs of security for the resulting crypto systems. Our techniques have a number of crucial advantages.

They are provably secure, they provide provable secrecy for encryption, in the sense that the untrusted server cannot learn anything about the plaintext when only given the cipher text, they provide query isolation for searches, meaning that the untrusted server cannot learn anything more about the plaintext than the search result; they provide controlled searching, so that the untrusted server cannot search for an arbitrary word without the user's authorization; they also support hidden queries, so that the user may ask the untrusted server to search for a secret word without revealing the word to the server. The algorithms presented are simple, fast (for a document of length n, the encryption and search algorithms only need O(n) stream cipher and block cipher operations), and introduce almost no space and communication overhead, and hence are practical to use today

2.4 Searchable symmetric encryption: improved definitions and efficient constructions

AUTHORS: R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky

Searchable symmetric encryption (SSE) allows a party to outsource the storage of its data to another party (a server) in a private manner, while maintaining the ability to selectively search over it. This problem has been the focus of active research in recent years. In this paper we show two solutions to SSE that simultaneously enjoys the following properties,

Both solutions are more efficient than all previous constant-round schemes. In particular, the work performed by the server per returned document is constant as opposed to linear in the size of the data.

Both solutions enjoy stronger security guarantees than previous constant-round schemes. In fact, we point out subtle but serious problems with previous notions of security for SSE, and show how to design constructions that avoid these pitfalls.

Further, our second solution also achieves what we call adaptive SSE security, where queries to the server can be chosen adaptively (by the adversary) during the execution of the search; this notion is both important in practice and has not been previously considered.

Surprisingly, despite being more secure and more efficient, our SSE schemes are remarkably simple. We consider the simplicity of both solutions as an important step towards the deployment of SSE technologies. As an additional contribution, we also consider multi-user SSE. All prior work on SSE studied the setting where only the owner of the data is capable of submitting search queries. We consider the natural extension where an arbitrary group of parties other than the owner can submit search queries. We formally define SSE in the multi-user setting, and present an efficient construction that achieves better performance than simply using access control mechanisms.

III. PROPOSED SYSTEM

In this paper, we propose PRMSM, a privacy preserving ranked multi-keyword search protocol in a multi-owner cloud model. We define a multi-owner model for privacy preserving keyword search over encrypted cloud data. We propose an efficient data user authentication protocol, which not only prevents attackers from eavesdropping secret keys and pretending to be illegal data users performing searches, but also enables data user authentication and revocation.

We systematically construct a novel secure search protocol, which not only enables the cloud server to perform secure ranked keyword search without knowing the actual data of both keywords and trapdoors, but also allows data owners to encrypt keywords with self-chosen keys and allows authenticated data users to query without knowing these keys. We propose an Additive Order and Privacy Preserving Function family (AOPPF) that allows data owners to protect the privacy of relevance scores using different functions according to their preference, while still permitting the cloud server to rank the data files accurately. We conduct extensive experiments on real-world datasets to confirm the efficacy and efficiency of our proposed schemes.

3.1 System architecture and implementation

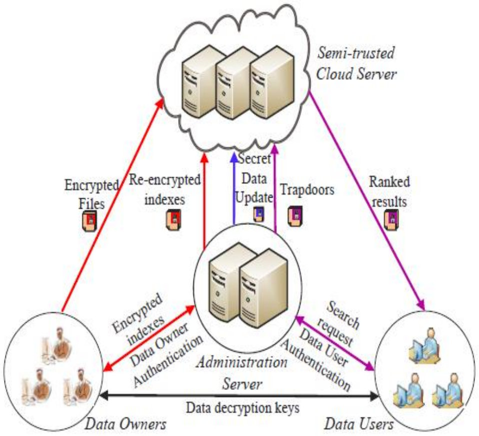

Figure 3: System Architecture

3.2 Implementation Modules

1. System Model
2. Data User Authentication
3. Illegal Search Detection
4. Search over Multi-owner

3.3 MODULES DESCRIPTION:

3.3.1 System Model

In the first module, we design and develop the System Model to implement our proposed system. Our System model consists of Admin, users, data owners, and Cloud Servers. Admin provides the accessibility to data-owners. Initially Data-owner has to register and admin approves the each data owner request. The respective Password and login credentials will be sent to the Email ID of Data owner. In Users sub-module, each user has a global identity in the system. A user may be entitled a set of attributes that may come from multiple attribute authorities. The user will receive a secret key associated with its attributes entitled by the analogous attribute dominance. In data owner's sub-module, the proposed scheme should allow new data owners to enter this system without affecting other data owners or data users, i.e., the plan should support data owner scalability in a plug-and-play model. In Cloud Server sub-module of system model, the owner sends the encrypted data to the cloud server through Admin. They do not rely on the server to do data access control. But, the access control happens inside the cryptography.

3.3.2 Data User Authentication

To protect user data from attackers and pretending to be legal data users performing searches and launching statistical attacks based on the search result, data users must be authenticated before the administration server re-encrypts trapdoors for data users. Traditional authentication methods often follow three steps. First, data requester and data authenticator share a secret key, say, $k0$. Second, the requester encrypts his personally identifiable information $d0$ using $k0$ and sends the encrypted data $(d0)k0$ to the authenticator. Third, the authenticator decrypts the received data with $k0$ and authenticates the decrypted data. To accomplish desired successful authentication is to provide both the dynamically changing secret keys and the historical data of the corresponding data user.

3.3.3 Illegal Search Detection

In search detection module scheme, the dynamic secret key and the historical information protect the authentication process. We assume that an attacker has successfully eaves dropped the secret key. Then attacker has to construct the authentication data, if the attacker has not successfully eavesdrops the historical data, e.g., the request counter, the last request time, he cannot design the correct authentication data. Therefore the administration server will soon detect this illegal trapdoors action. Further, if the attacker has successfully eavesdropped all data of us*ers*. The attacker can correctly design the authentication data and pretend him as u*ser* without being detected by the administration server. However, once the user perform legal data search, since the secret key on the administration server side has changed, there will be contradictory secret keys between the administration server and the authenticated data user. Therefore, the data user and administration server will soon detect this illegal action.

3.3.4 Search over Multi-owner:

The proposed scheme should allow multi-keyword search over encrypted files that would be encrypted with different keys for different data owners. It also needs to allow the cloud server to rank the search results among different data owners and return the top-*k* results. The cloud server stores all encrypted files and keywords of different data owners. The administration server will also store a secret data on the cloud server. Upon receiving a query request, the cloud will search over the data of all these data owners. The cloud processes the search request in two steps. First, the cloud matches the queried keywords from all keywords stored on it, and it gets a candidate file set. Second, the cloud ranks files in the candidate file set and finds the most top-*k* relevant files. Finally, we apply the proposed scheme to encode the relevance scores and obtain the top-*k* search results.

IV. ALGORITHM

AES is an iterative rather than Feistel cipher. It is based on 'substitution–permutation network'. The AES compares a series of inter linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations). Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix. Unlike DES, the number of rounds in AES is variable and depends on the length of the key. Advance Encryption Standard (AES) uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

The Advance encryption standard (AES) schematic structure is given in figure 4.

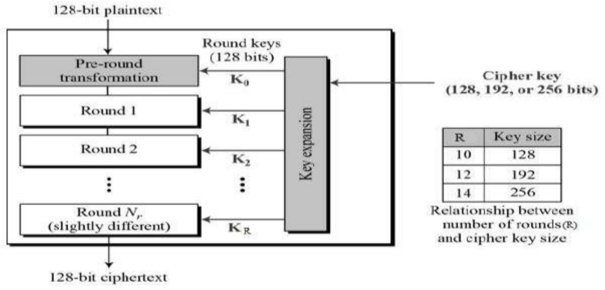

Figure 4: Encryption Process

Here, we restrict to description of a typical round of AES encryption. Each round comprise of four sub-processes. The first round process is depicted in figure 5.

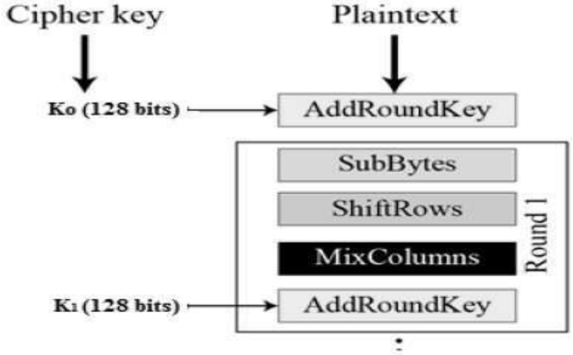

Figure 5: Round Process

1. Byte Substitution (SubBytes)

In the byte substitution the 16 byte input data is substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns.

2. Shiftrows

Each of the four rows of the matrix is shifted to the left. Any entries that 'fall off' are re-inserted on the right side of row. Shift is carried out as follows,

- First row is not shifted.
- Second row is shifted one (byte) position to the left.
- Third row of the matrix is shifted two positions to the left.
- Fourth row is shifted three positions to the left.
- The result after shift row is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

3. MixColumns

Each column of four bytes is now transformed using a special mathematical function. As input the four bytes of one column and outputs four completely new bytes, which replace the original column by using the mathematical function. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.

4. Addroundkey

The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key. In the addroundkey this is the last round then the output is the cipher text. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

5. Decryption Process

The process of decryption of an AES cipher text is similar to the encryption process in the reverse order. Each round consists of the four processes conducted in the reverse order,

- Add round key
- Mix columns
- Shift rows
- Byte substitution

Since sub-processes in each round are in reverse manner, unlike for a Feistel Cipher, the encryption and decryption algorithms need to be separately implemented, although they are very closely related.

V. CONCLUSION

In this paper, we explore the problem of secure multi-keyword search for multiple data owners and multiple data users in the cloud-computing environment. Different from prior works, our schemes enable authenticated data users to achieve secure, convenient, and efficient searches over multiple data owners' data. To efficiently authenticate data users and detect attackers who steal the secret key and perform illegal searches, we propose a novel dynamic secret key generation protocol and a new data user authentication protocol. To enable the cloud server to perform secure search among multiple owners' data encrypted with different secret keys, we systematically construct a novel secure search protocol. The search results are ranked and preserve the privacy of relevance scores between keywords and files. We propose a novel Additive Order and Privacy Preserving Function family. Moreover, we show that our approach is computationally efficient, even for large data and keyword sets. The future work, on one hand, we will consider the problem of secure fuzzy keyword search in a multi-owner paradigm. On the other hand, we plan to implement our scheme on the commercial clouds.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communication of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacypreserving public auditing for secure cloud storage," *Computers, IEEE Transactions on*, vol. 62, no. 2, pp. 362–375, 2013.

[3] D.Song, D.Wagner, and A.Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE International Symposium on Security and Privacy (S&P'00)*, Nagoya, Japan, Jan. 2000, pp. 44–55.

[4] E. Goh. (2003) Secure indexes. [Online]. Available: http://eprint.iacr.org/

[5] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in *Proc. ACM CCS'06*, VA, USA, Oct. 2006, pp. 79–88.

[6] D. B. et al., "Public key encryption with keyword search secure against keyword guessing attacks without random oracle," *EUROCRYPT*, vol. 43, pp. 506–522, 2004.
[7] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in *Proc. Applied Cryptography and Network Security (ACNS'04)*, Yellow Mountain, China, Jun. 2004, pp. 31–45.
[8] L. Ballard, S. Kamara, and F. Monrose, "Achieving efficient conjunctive keyword searches over encrypted data," in *Proc. Information and Communications Security (ICICS'05)*, Beijing, China, Dec. 2005, pp. 414–426.
[9] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in *Proc. IEEE Distributed Computing Systems (ICDCS'10)*, Genoa, Italy, Jun. 2010, pp. 253–262.
[10] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacypreserving multi-keyword ranked search over encrypted cloud data," in *Proc. IEEE INFOCOM'11*, Shanghai, China, Apr. 2011, pp. 829–837.