ISSN 2277-2685
IJESR/February. 2019/ Vol-9/Issue-1/133-139

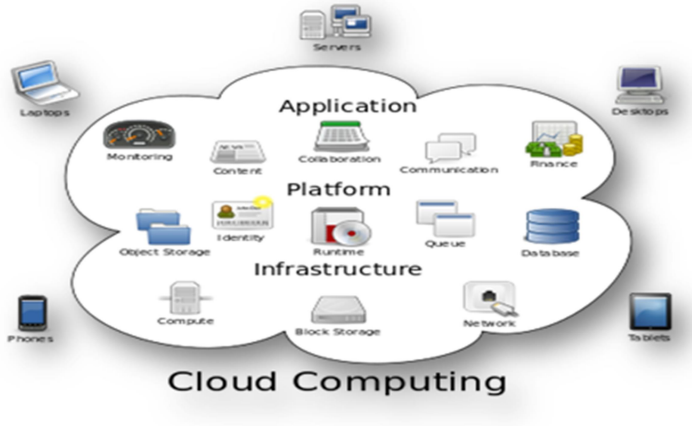
SECURE CLOUD STORAGE WITH KEYWORD SEARCH AND DUAL-SERVER PUBLIC-KEY ENCRYPTION

Dr. G MANIGANTA[*1], B. VENKATESWARA RAO[*2], B. PREM KUMAR[*3]
[*1] **Associate Professor, Dept. of Computer Science Engineering,**
[*2, 3] ***Assistant Professor, Dept. of Computer Science Engineering.***
A.M Reddy Memorial College of Engineering and Technology, Andhra Pradesh

Abstract: A growing number of individuals are turning to searchable encryption to safeguard the privacy of their data in secure cloud storage environments. In this study, we investigate the security of public key encryption with keyword search (PEKS), a fundamental cryptographic technique widely utilized in cloud storage applications. Unfortunately, the traditional PEKS framework has been found to suffer from a vulnerability known as an inside keyword guessing attack (KGA), which can be exploited by a malicious server. To address this security flaw, we propose the dual-server PEKS framework (DS-PEKS) as a novel approach. Additionally, we introduce a new type of smooth projective hash function (SPHF) called a linear and homomorphic SPHF (LH-SPHF). Leveraging LH-SPHF, we present a generic construction of secure DS-PEKS. Furthermore, we offer an effective instantiation of this general framework based on Decision Diffie-Hellman LH-SPHF, demonstrating its ability to achieve robust security against inside KGA attacks.
Indexed Terms -- Location-based social network, text mining, travel route recommendation

I. INTRODUCTION

Cloud computing (equipment and software) is used and shared remotely over a network in what is known as "the cloud" (usually the Internet). In structure graphs, a cloud-shaped picture is commonly used to represent the complex information it contains, hence the name. Through distributed processing, a client's information, code, and estimation can be shared amongst multiple, geographically dispersed organizations. System hardware and software for appropriate processing are available online from supervised pariah groups. Modern programming languages and server PC networks are made possible by these establishments.



Structure of cloud computing

Explaining the Workings of Cloud Computing.

Traditional supercomputing, or peak execution handling power, is typically reserved for use by the military and assessment agencies. The purpose of distributed registration is to put this type of processing power to use in client-centric applications, such as financial portfolios, the transmission of updated information, the provision of data limits, and the management of massive, visually impressive PC games. Distributed processing makes use of networks of very large groups of servers, which typically run low-cost client PC development and have some connection to dispersing data-handling tasks. Common IT architectures feature massive aggregations of interconnected systems. Virtualization methods are commonly used to increase the efficiency of distributed computing. Characteristics and Service Types: With the NIST's definitions in mind, here are some of the most remarkable aspects of widely disseminated numbers:

- Self-organization on demand: customer can set their limits for things like server time and association storage as needed, without needing to coordinate with each specialist facility individually.
- Capabilities are accessible over the network and can be used by a variety of client types thanks to standardized frameworks (e.g., cells, PCs, and PDAs).
- Resource pooling: In a multi-tenant model, the provider shares its enlisting resources among its many clients, allocating and reallocating its physical and digital assets to each client by their needs. Since the client generally has no control or data over the specific region of the provided resources at this point, there is a sense of region opportunity and the client may have the option to decide region at a higher level of reflection (e.g., country, state, or server ranch). Resource situations consist of constraints, management, memory, data transmission over networks, and virtual machines.
- Rapid adaptability: Capabilities can be provisioned quickly and skillfully, occasionally normally, to rapidly scale out, and immediately conveyed to rapidly scale in. Often, the client has the impression that they can purchase an unlimited amount of provisioning at any time.
- A metering limit appropriate to the type of business is typically used by cloud architectures to manage and expand resource utilization (e.g., limit, dealing with, information transmission, and dynamic client accounts). Both the user and the resource provider can benefit from due, controlled, and definitive resource use.

Characteristics of cloud computing

II. RELATED WORK

1) A new generic framework with a keyword search for safe public key encryption
R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang are the authors.
Users can search encrypted files on an untrusted server using Public Key Encryption with Keyword Search (PEKS), a technology developed by Boneh et al. in Eurocrypt'04. The cryptography research community has paid a lot of attention to this idea because it has many practical uses. All of the current PEKS schemes, however, have the drawback of being unable to withstand the Keyword Guessing Attack (KGA) initiated by a hostile server. In this research, we present Dual-Server Public Key Encryption with Keyword Search as a new PEKS architecture (DS-PEKS). As long as the two untrusted servers do not cooperate, this new structure can withstand every assault, including the KGA. Then, using a fresh iteration of the Smooth Projective Hash Functions (SPHFs), we propose a general construction of DS-PEKS that is also of interest.
2) Improved definitions and effective structures for searchable symmetric encryption
R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky are the authors.
A party can delegate the private storage of his data to a third party while still having the option to conduct limited searches on it thanks to searchable symmetric encryption (SSE). Research on this issue has been ongoing, and several security definitions and constructions have been put out. In this essay, we first examine current security concepts before putting forth new, more robust security definitions. Then, we provide two constructions that, according to our new definitions, are secure. Interestingly, our buildings are more effective than any preceding constructions and also satisfy better security assurances.
Additionally, earlier research on SSE only took into account the scenario in which only the owner of the data can submit search queries. We take into account the logical extension where search queries can be submitted by any arbitrary group of parties aside from the owner. In this multi-user environment, we explicitly define SSE and propose an effective construction.
3) K-Resilient IBE-based Public Key Encryption with Keyword Search
TITLE: D. Khader
Abstract. Bob sends Alice an email that has been encrypted. For some reason, a gateway needs to see if a specific keyword is present in an email or not (e.g. routing). However, Alice does not want anybody else, not even the gateway, to be able to decode the email. This situation calls for the use of public key encryption with keyword search (PEKS). In this study, we develop the KResilient Public Key Encryption with Keyword Search (KR-PEKS), a novel technique. Without the random oracle, the new technique is secure against a chosen keyword attack. The
The KR-PEKS was created using the capability of creating a Public Key Encryption with Keyword Search from an Identity Based Encryption. By demonstrating that the used IBE had a notion of key privacy, the security of the proposed system was demonstrated. The system was then changed in two distinct ways to achieve each of the following: the first change enabled multiple keyword searches, and the second change did away with the requirement for secure channels.
4) Generic secure-channel encryption that is open to search and has adaptive security
K. Emura, A. Miyaji, M. S. Rahman, and K. Omote are the authors.
A public key encryption system with keyword search (PEKS) and its variant secure-channel free PEKS (SCF-PEKS) have been proposed for keyword searches against encrypted material. In this research, we expand the security of SCF-PEKS and provide adaptive SCF-PEKS, where an adversary is allowed to issue test queries adaptively (modeled as a "malicious but legitimate" receiver). We demonstrate that only anonymous identity-based encryption is capable of generically constructing adaptive SCF-PEKS. In contrast to the PEKS construction by Abdalla et al. (2008), SCF-PEKS can be created without the need for any additional cryptographic primitives, even though adaptive SCF-PEKS necessitates additional capabilities.

We also provide an alternative adaptive SCF-PEKS structure that is more effective than the previous one while not being entirely generic. In comparison to the (non-adaptive secure) SCF-PEKS scheme by Fang et al., we finally instantiate an adaptive SCF-PEKS scheme (using our second construction) that achieves a similar degree of efficiency for the costs of the test procedure and encryption (CANS2009). 2014 John Wiley & Sons, Ltd. Copyright 5) Cooperative data possession for multi-cloud storage integrity verification

5) Offline keyword guessing attacks using keyword search techniques on modern public key encryption

W.-C. Yau, S.-H. Heng, and B.-M. Goi is the author.

Boneh et al. introduced the Public Key Encryption with Keyword Search Scheme (PEKS) for the first time in 2004. The issue of searching through material that has been encrypted with a public key setting is resolved by this scheme. The Secure Channel Free Public Key Encryption with Keyword Search (SCF-PEKS) technique, which eliminates the secure channel for sending trapdoors, was recently suggested by Baek et al. Later on, they put out the PKE/PEKS system, an upgraded PEKS method that incorporates a PKE scheme. We discuss offline keyword guessing attacks against SCF-PEKS and PKE/PEKS methods in this work. We show that offline keyword guessing attacks allow external adversaries who intercept trapdoors sent over a public channel to decrypt encrypted keywords. While internal enemies can carry out assaults regardless of whether trapdoors are supplied over a secure or public channel.

III. SYSTEM ANALYSIS

Existing System

- In a PEKS system, the sender encrypts some keywords (called PEKS ciphertexts) with the receiver's public key and appends them to the encrypted data. The receiver then communicates with the server with the backdoor of a search term. The server can determine if the keyword used by the receiver in the PEKS ciphertext is the same as the one used in the trapdoor by comparing the two. If it does, the server will give the recipient the appropriate encrypted information. A new PEKS strategy, called a secure channel-free PEKS, was proposed by Baek et al., which eliminates the need for a secure channel (SCF-PEKS).
- Security for SCF-PEKS was improved by Rhee et al. after it was discovered that an attacker may learn about the connection between the non-challenge ciphertexts and the trapdoor.
- Since users typically employ well-known terms when looking for documents, Byun et al. introduced the offline keyword guessing attack against PEKS.

CONS: The current system has many drawbacks.

- Despite not requiring the dissemination of secret keys, PEKS methods are not completely secure due to a vulnerability in the trapdoor keyword privacy, more specifically the Keyword Guessing Attack (KGA). Security is compromised because anyone with knowledge of the receiver's public key can produce the PEKS ciphertext of any random keyword.
- To be more precise, an adversarial server equipped with a trapdoor can select a guessing term from the keyword space and use it to produce a PEKS ciphertext. When a guess is made, the server can see if it matches the secret keyword. Repeating this process of guessing and testing until the right keyword is identified is possible.
- One problem is that the server does not have a hard time determining which small set the underlying keyword is a part of, even if it cannot guess the keyword itself. This means that

the keyword's privacy is not properly safeguarded from the server. However, their plan is infeasible since the recipient must independently locate the correct ciphertext by utilizing the exact trapdoor to eliminate all but the one correct answer from the set supplied by the server.

THE SUGGESTED SYSTEM:

This study makes four major contributions.

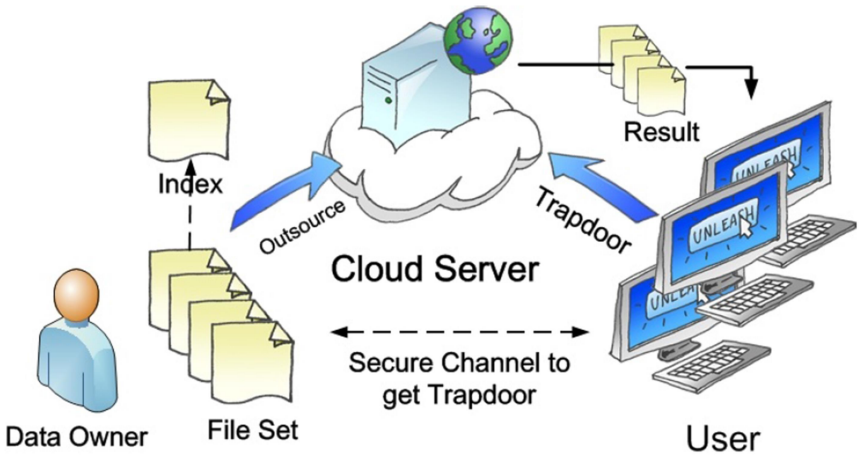
- To fix this security hole in PEKS, we create a new framework we call Dual-Server Public Key Encryption with Keyword Search (DS-PEKS).
- For a more general DS-PEKS design, we present a linear and homomorphic form of the Smooth Projective Hash Function (SPHF) called linear SPHF.
- Using the proposed Lin-Hom SPHF, we demonstrate a generic construction of DS-PEKS.
- In this study, we describe a practical implementation of our SPHF based on the Diffie-Hellman language to demonstrate the viability of our novel framework.

PROPOSED SYSTEM BENEFITS:

- Since our technique does not require any pairing computation, it is more efficient than existing systems that do require pairing computation during the production of PEKS ciphertext and testing.
- It has been determined that our method achieves the highest PEKS computation efficiency. The reason is, our plan leaves out the computation of pairs deliberately. In particular, the current technology has the highest calculation cost because each generation of PEKS requires 2 pairing computations.
- We do not need to perform any pairing computation, and the server takes care of all the searching, so our scheme has a lower computation cost than any existing scheme, despite requiring an additional step for the testing.

IV. SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE:



MODULES:

- Module for Building Systems

- Security against Selective Keyword Semantics
- Server at the Front
- Back Server

CONTENTS OF MODULES:

Module for Building Systems

Our first module focuses on building the foundation of our system by creating the necessary data structures and entities. 1) Cloud User: the person or business who initially placed data in the cloud and is now using that data. 2) CSP: cloud service provider, the company in charge of running CSS and selling access to their network's cloud storage space to customers. We introduce the formal concept and security models of the new framework we propose, called DS-PEKS. Later, a new type of smooth projective hash function is defined by us (SPHF). Formal correctness analysis and security proofs are presented to demonstrate a general construction of DS-PEKS from LH-SPHF. In conclusion, we show that SPHF can be used to efficiently implement DS-PEKS.

Security against Selective Keyword Semantics

To ensure that an attacker cannot tell one keyword apart from another when presented with a PEKS ciphertext, we implement semantic security against a chosen keyword attack in this section. In other words, an opponent cannot deduce the underlying keyword from the PEKS ciphertext.

Server at the Front

When the front server receives a query from the receiver, it uses its private key to pre-process the trapdoor and all the PEKS ciphertexts. It then provides the rear server with a set of testing states that conceal the relevant trapdoor and PEKS ciphertexts.

Back Server

Using its private key and the obtained internal testing states from the front server, the back server can then decide which documents are queried by the receiver in this section.

V. CONCLUSION

In this research, we propose a novel framework called Dual-Server Public Key Encryption with Keyword Search (DS-PEKS) to address a flaw in the standard PEKS architecture: the possibility of an inside keyword guessing attack. In addition, we developed a new Smooth Projective Hash Function (SPHF) and applied it to the creation of a generalized DS-PEKS protocol. The research also presents a practical implementation of the new SPHF based on the Diffie-Hellman issue, which yields a practical DS-PEKS method without pairings.

VI. FUTURE WORK

In the Future, Long-Term Impact: In Message Locked Encryption (MLE), a new cryptographic primitive, the key used to encrypt and decode the message is itself obtained from the message. Numerous cloud-storage companies aim to provide secure deduplication (space-efficient secure outsourced storage), and MLE gives the means to do so. It defines confidentiality as well as a type of integrity known as tag consistency.

.

VII. REFERENCES

[1] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, “A new general framework for secure public key encryption with keyword search,” in Proc. 20th Australasian Conf. Inf. Secur. Privacy (ACISP), 2015, pp. 59–76.

[2] D. X. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in Proc. IEEE Symp. Secur. Privacy, May 2000, pp. 44–55.

[3] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2004, pp. 563–574.
[4] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," in Proc. 13th ACM Conf. Comput. Commun. Secur. (CCS), 2006, pp. 79–88.
[5] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Proc. Int. Conf. EUROCRYPT, 2004, pp. 506–522.
[6] R. Gennaro and Y. Lindell, "A framework for password-based authenticated key exchange," in Proc. Int. Conf. EUROCRYPT, 2003, pp. 524–543.
[7] B. R. Waters, D. Balfanz, G. Durfee, and D. K. Smetters, "Building an encrypted and searchable audit log," in Proc. NDSS, 2004, pp. 1–11.
[8] M. Abdalla et al., "Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions," in Proc. 25th Annu. Int. Conf. CRYPTO, 2005, pp. 205–222.
[9] D. Khader, "Public key encryption with keyword search based on K-resilient IBE," in Proc. Int. Conf. Comput. Sci. Appl. (ICCSA), 2006, pp. 298–308.
[10] P. Xu, H. Jin, Q. Wu, and W. Wang, "Publickey encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack," IEEE Trans. Comput., vol. 62, no. 11, pp. 2266–2277, Nov. 2013.