**Dual-Server Public-Key Encryption With Keyword Search for Secure Cloud Storage**

**ABSTRACT:**

Searchable encryption is of increasing interest for protecting the data privacy in secure searchable cloud storage. In this paper, we investigate the security of a well-known cryptographic primitive, namely, public key encryption with keyword search (PEKS) which is very useful in many applications of cloud storage. Unfortunately, it has been shown that the traditional PEKS framework suffers from an inherent insecurity called inside keyword guessing attack (KGA) launched by the malicious server. To address this security vulnerability, we propose a new PEKS framework named dual-server PEKS (DS-PEKS). As another main contribution, we define a new variant of the smooth projective hash functions (SPHFs) referred to as linear and homomorphic SPHF (LH-SPHF). We then show a generic construction of secure DS-PEKS from LH-SPHF. To illustrate the feasibility of our new framework, we provide an efficient instantiation of the general framework from a Decision Diffie–Hellman-based LH-SPHF and show that it can achieve the strong security against inside the KGA.

**EXISTING SYSTEM:**

* In a PEKS system, using the receiver’s public key, the sender attaches some encrypted keywords (referred to as PEKS ciphertexts) with the encrypted data. The receiver then sends the trapdoor of a to-be-searched keyword to the server for data searching. Given the trapdoor and the PEKS ciphertext, the server can test whether the keyword underlying the PEKS ciphertxt is equal to the one selected by the receiver. If so, the server sends the matching encrypted data to the receiver.
* Baek *et al.* proposed a ew PEKS scheme without requiring a secure channel, which is referred to as a secure channel-free PEKS (SCF-PEKS).
* Rhee *et al.* later enhanced Baek *et al*.’s security model for SCF-PEKS where the attacker is allowed to obtain the relationship between the non-challenge ciphertexts and the trapdoor.
* Byun *et al.*introduced the off-line keyword guessing attack against PEKS as keywords are chosen from a much smaller space than passwords and users usually use well-known keywords for searching documents.

**DISADVANTAGES OF EXISTING SYSTEM:**

* Despite of being free from secret key distribution, PEKS schemes suffer from an inherent insecurity regarding the trapdoor keyword privacy, namely *inside Keyword Guessing Attack* (KGA). The reason leading to such a security vulnerability is that anyone who knows receiver’s public key can generate the PEKS ciphertext of arbitrary keyword himself.
* Specifically, given a trapdoor, the adversarial server can choose a guessing keyword from the keyword space and then use the keyword to generate a PEKS ciphertext. The server then can test whether the guessing keyword is the one underlying the trapdoor. This *guessing-then-testing* procedure can be repeated until the correct keyword is found.
* On one hand, although the server cannot exactly guess the keyword, it is still able to know which small set the underlying keyword belongs to and thus the keyword privacy is not well preserved from the server. On the other hand, their scheme is impractical as the receiver has to locally find the matching ciphertext by using the exact trapdoor to filter out the non-matching ones from the set returned from the server.

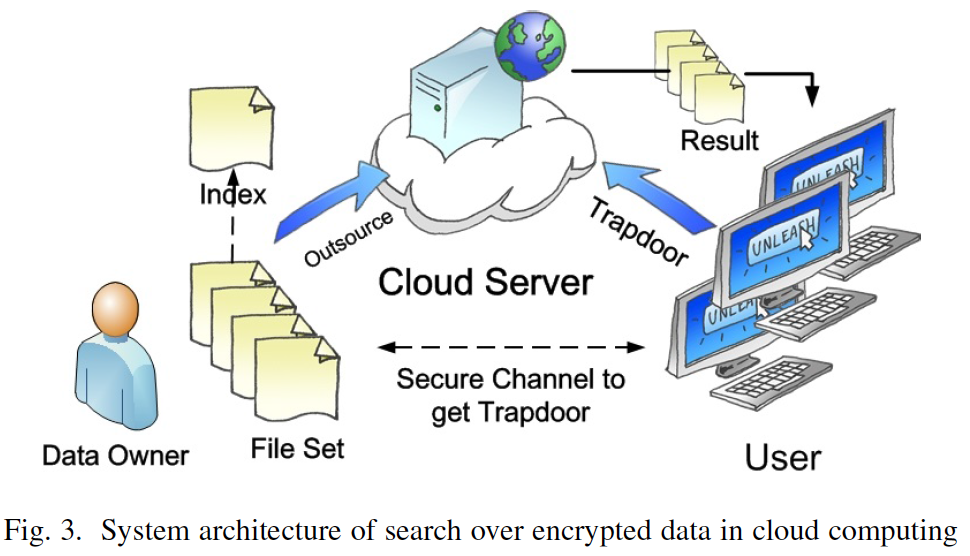
**PROPOSED SYSTEM:**

* The contributions of this paper are four-fold.
* We formalize a new PEKS framework named *Dual-Server Public Key Encryption with Keyword Search* (DS-PEKS) to address the security vulnerability of PEKS.
* A new variant of *Smooth Projective Hash Function* (SPHF), referred to as *linear and homomorphic SPHF*, is introduced for a generic construction of DS-PEKS.
* We show a generic construction of DS-PEKS using the proposed Lin-Hom SPHF.
* To illustrate the feasibility of our new framework, an efficient instantiation of our SPHF based on the Diffie-Hellman language is presented in this paper.

**ADVANTAGES OF PROPOSED SYSTEM:**

* All the existing schemes require the pairing computation during the generation of PEKS ciphertext and testing and hence are less efficient than our scheme, which does not need any pairing computation.
* Our scheme is the most efficient in terms of PEKS computation. It is because that our scheme does not include pairing computation. Particularly, the existing scheme requires the most computation cost due to 2 pairing computation per PEKS generation.
* In our scheme, although we also require another stage for the testing, our computation cost is actually lower than that of any existing scheme as we do not require any pairing computation and all the searching work is handled by the server.

**SYSTEM ARCHITECTURE:**



**MODULES:**

* System Construction Module
* Semantic-Security against Chosen Keyword Attack
* Front Server
* Back Server

**MODULES DESCSRIPTION:**

**System Construction Module**

In the first module, we develop the system with the entities required to provde our system. 1) Cloud User: the user, who can be an individual or an organization originally storing their data in cloud and accessing the data. 2) Cloud Service Provider (CSP): the CSP, who manages cloud servers (CSs) and provides a paid storage space on its infrastructure to users as a service. We propose a new framework, namely DS-PEKS, and present its formal definition and security models. We then define a new variant of smooth projective hash function (SPHF). A generic construction of DS-PEKS from LH-SPHF is shown with formal correctness analysis and security proofs. Finally, we present an efficient instantiation of DS-PEKS from SPHF.

**Semantic-Security against Chosen Keyword Attack**

In the module, we develop the semantic-security against chosen keyword attack which guarantees that no adversary is able to distinguish a keyword from another one given the corresponding PEKS ciphertext. That is, the PEKS ciphertext does not reveal any information about the underlying keyword to any adversary.

**Front Server:**

After receiving the query from the receiver, the front server pre-processes the trapdoor and all the PEKS ciphertexts using its private key, and then sends some internal testing-states to the back server with the corresponding trapdoor and PEKS ciphertexts hidden.

**Back Server:**

In this module, the back server can then decide which documents are queried by the receiver using its private key and the received internal testing-states from the front server.

**SYSTEM REQUIREMENTS:**

**HARDWARE REQUIREMENTS:**

* System : Pentium Dual Core.
* Hard Disk : 120 GB.
* Monitor : 15’’ LED
* Input Devices : Keyboard, Mouse
* Ram : 1GB.

**SOFTWARE REQUIREMENTS:**

* Operating system : Windows 7.
* Coding Language : JAVA/J2EE
* Tool : Netbeans 7.2.1
* Database : MYSQL

**REFERENCE:**

Rongmao Chen, Yi Mu, *Senior Member, IEEE*, Guomin Yang, *Member, IEEE*, Fuchun Guo, and Xiaofen Wang, “Dual-Server Public-Key Encryption With Keyword Search for Secure Cloud Storage”, **IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 11, NO. 4, APRIL 2016.**