Implementation of Fast keyword search using Public-Key Ciphertexts with Hidden Structures (FKSCH)

Srinath K S[#1], GovindRaj J[#2], Kirti Kumari [#3], Poojitha. M[#4], Pintu Kumar [#5]
[#]*CS&E department, Sambhram Institute of Techonology*
Bengaluru-97, Karnataka, India
[1]*solansrinath@gmail.com*
[2]*govindraj7695@gmail.com*
[3]*kirtikumari0073@gmail.com*
[4]*poojitha96reddy@gmail.com*
[5]*pintujaadav@gmail.com*

Abstract— **In today's world, various methods are being deployed to establish a secure communication, fast keyword searching, and retrieval completeness over a network. The existing semantic secure Public-key Encryption with Keyword Search (PEKS) takes linear search time with the total number of cipher texts which makes it time consuming. The retrieval of keywords from the large-scale database is complex and restricted. To tackle with this drawback, our paper proposes complete system that implements a Fast keyword search using Public-Key Ciphertexts with Hidden Structures (FKSCH) for fast keyword search without sacrificing the linguistic security of encrypted keywords. The searchable cipher text keywords are structured by hidden relations, which reveal only a part of information to match the cipher text efficiently by preserving the privacy of the data. Hence the time taken by SPCHS for the keyword search depends on actual number of cipher text containing queried keyword rather than the whole cipher text in the cloud database. Our scheme satisfies the standard rules of Random Oracle model and proves to be better than standard model.**

Keywords— **PEKS, FKSCH, Public-key, Hidden structure.**

I. Introduction

Cloud computing is Internet-based computing where shared resources, software, and information are provided to computers and other devices on demand. Huge sensitive data are encrypted using public key encryption technique and stored in a cloud database. The Public Key Encryption along with Keyword Search allows one to search the data that is in encrypted form with a keyword without showing any information. This scheme gives the detail study on implementation of Fast keyword search using Public-Key Ciphertexts with Hidden Structures (FKSCH) that fasten the keyword search without sacrificing the security of encrypted keywords. In FKSCH, the keyword ciphertexts are structured by hidden relation and by using a trapdoor function which disclose minimum information to search algorithm hence privacy is maintained. All keyword searchable ciphertexts are structured by hidden relations, and with trapdoor corresponding to a keyword, search algorithm ushers to find all matching ciphertexts efficiently faster. Search complexity depends on the actual number of ciphertexts containing queried keywords rather than the number of all ciphertexts.

FKSCH is constructed by using Identity – Based Encryption (IBE) and collision-free full-identity malleable from scratch in which ciphertexts contains a hidden star like structure to provide security to sensitive data stored in cloud database as well as to search these files faster and it semantically satisfies with standards of Random Oracle (RO) Model.

II. Related Work

In current era, the increasing demand of cloud usage leads to do lot of research on searching encrypted data in cloud database. It spots a light on two general categories such as public-key searchable encryption and symmetric encryption.

PUBLIC-KEY encryption with keyword search (PEKS), introduced by Boneh et al., In which anyone who knows the receiver's public key can upload keyword-searchable ciphertexts to a cloud. The receivers can do keyword search in the server. The drawback of this idea is, it take search time liner with total number of all ciphertexts present in cloud database. This makes retrieval from certain database prohibitive [1].

M. Bellare, A. Boldyreva and A. O'Neill, presented as-strong-as-possible definitions of privacy and constructing them for achieving public-key encryption schemes using deterministic encryption and searchable algorithm. It indicates the database encryption method that permit fast search while provably providing privacy and fast search constraint. The main advantage of this method is One can efficiently encrypt long messages using RSA-DOAEP (deterministic optimal asymmetric encryption padding) without making use of hybrid encryption and ESE scheme provides fast search. The drawback of this model is it configure to single user setting hence it as one receiver and one public key [2].

Boneh and Boyen developed two efficient IBE systems that are provably selective identity secure without the random oracle model. In these systems, encryption requires no bilinear map computation and decryption requires at most two. The first method is based on the Decision Bilinear Diffie-Hellman (Decision BDH) assumption. This construction extends to give an efficient selective identity secure hierarchical IBE (HIBE) without random oracles. Second method is even more efficient, but it is based on a non-standard assumption called as Decision Bilinear Diffie-Hellman Inversion (Decision BDHI) [3]. Their idea eliminates the use of random oracles and also Reduced the size of cipher text, But it is Highly impractical [3].

Boyen and Waters presented an identity based cryptosystem that features fully anonymous ciphertext and hierarchical key delegation. The cryptographic primitive of

Identity-Based Encryption allows a sender to encrypt a message for a receiver using only the receiver's identity as a public key in "anonymous" identity-based encryption systems, where the ciphertext does not leak the identity of the recipient. Anonymous IBE systems can be leveraged to construct Public Key Encryption with Keyword Search (PEKS) schemes. Anonymous HIBE further enables sophisticated access policies for PEKS and ID-based PEKS. Existing systems don't have both anonymous and without random oracle property [4].

Huang H., Yang B, proposed a new ElGamal public key encryption scheme based on a new Diffie-Hellman problem called EDDH problem. ElGamal System presents a public key cryptosystem based on the Discrete-log problem. This scheme is known as ElGamal cryptosystem, it modifies the Diffie-Hellman protocol with the goal so that it can be used as an encryption and decryption protocol. Its security is also based on the difficulty of the DLP(Discrete Logarithm Problem). The security of both systems depends on the trouble of figuring discrete logarithms over finite fields. To secure against mathematical and brute-force attack as well as Low-Modulus and Known-Plaintext attack on ElGamal, they go for adjusted ElGamal cryptosystem algorithm. One of the strength of ElGamal is its non-determinism-encrypting the same plaintext multiple times will result in different ciphertexts, since a random k is chosen each time. It always needs for randomness, and its slower speed for purpose of signing. The potential disadvantage of the ElGamal system is that message expansion by a factor of two takes place during encryption [5].

III. Problem Statement

The current system of PEKS (Public Key Encryption along with Keyword Search) takes linear search time with total number of ciphertexts which restricts accessing of data from the huge database. PEKS scheme allows one to search the encrypted data with a keyword without revealing any information. Therefore, it needs more efficient search performance for deploying the PEKS scheme. To improve the search performance, deterministic encryption was introduced. But it also has two main limitations in maintaining privacy and information leaks [6]. To overcome from these problems we introduced implementation of Fast keyword search using Public-Key Ciphertexts with Hidden Structures (FKSCH) that proves to be semantically secure in the Random Oracle (RO) and standard models.

IV. Design Issues in FKSCH

Usage of Cloud computing has many drawbacks such as dependency, risk, security and searching. Our scheme mainly concentrates on fast ciphertexts keyword search and security in cloud. As cloud being a public service opens up cloud service providers a many security challenges. The attacks that can be performed on cloud are Denial-of-Service attack (DoS), Cloud Malware-Injection attack, Side Channel Attack, Authentication attack, and Man-in-the-middle cryptographic attacks and so on. The ease in procuring and accessing cloud services can also give active users the ability to scan, identify and exploit loopholes and vulnerabilities within a system. For instance, in a multi-tenant cloud architecture where multiple users are hosted on the same server, a hacker might try to break into the data of other users hosted and stored on the same server [7].

Searching a file on cloud becomes complex as the cloud is used for huge database. In our scheme, we provide a faster search of file using the trapdoor algorithm where the time complexity depends on the number of ciphertext containing the queried keyword rather than the total number of ciphertext hence, boost the search speed [8].

A. System Architecture without Trapdoor

In Public key Encryption with keyword search (PEKS), any user having receiver's public key can upload the encrypted file to the cloud database [9]. When a receiver want to Search particular file from server the search operation at server side is performed linearly and transferred to requested receivers. The architectural diagram Fig. 1 shows that the sender will encrypt the plain text file using the public key of receiver and generates a keyword searchable ciphertext. Sender uploads the encrypted file to the server and sends the keyword to the receiver, simultaneously. The Receiver delegates the keyword searchable ciphertexts to the server i.e., the server finds the encrypted files containing the queried keyword without knowing the original files or the keyword itself, and returns the corresponding encrypted files to the receiver. Finally, the receiver decrypts these encrypted files.

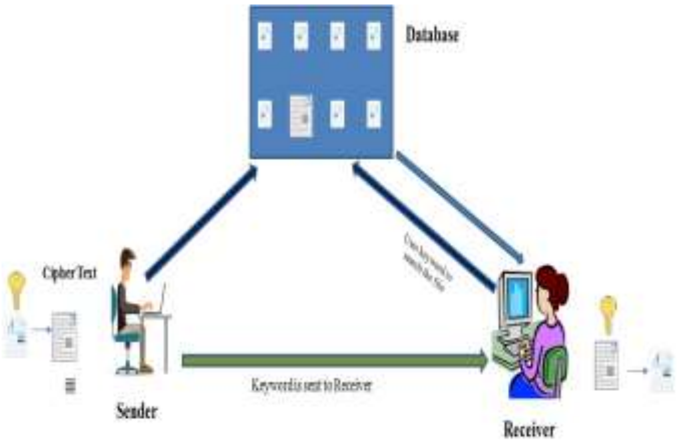


Fig. 1 Public key Encryption with keyword search

Existing semantically secure PEKS schemes take linear search time with the total number of all ciphertexts. This makes retrieval from large-scale databases prohibitive. Therefore, more efficient search performance is crucial for practically deploying PEKS schemes.

B. System Architecture with Trapdoor

In FKSCH, keyword searchable ciphertexts with their hidden structures can be generated in the public key setting, with a keyword search trapdoor, partial relations can be disclosed to guide the discovery of all matching ciphertexts. Semantic security is defined for both the keywords and the hidden structures [10].

The architecture of the proposed system described in detail in Fig. 2. The sender will encrypt the plain text file and generate keyword searchable ciphertext using the receiver public key. Sender uploads the encrypted file to the server and sends the keyword to the receiver, simultaneously. The Receiver delegates the keyword searchable ciphertexts to the server. The trapdoor algorithm is used at the server side for helping the cloud server to search the cipher text containing queried keyword faster with the help of hidden star like structure embedded in it. Hence, the time taken for searching the ciphertexts is lesser than the previous system.

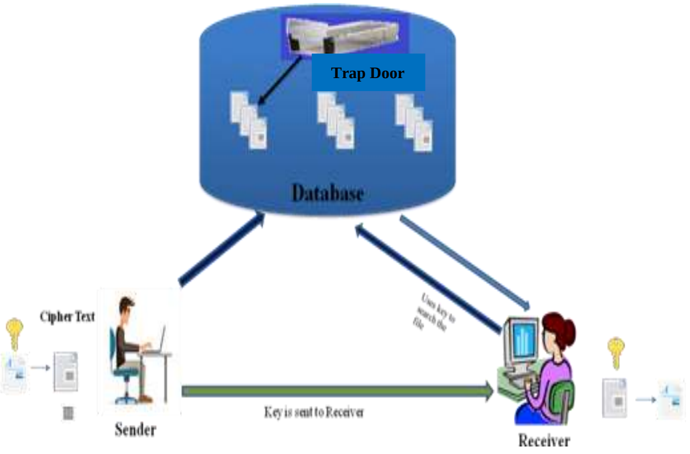


Fig. 1 Fast keyword search using Public-Key Ciphertexts with Hidden Structures

1) Hidden star like structure

The system has a target to increase search performance in FKSCH without sacrificing semantic security, where sender separately encrypts a file and its extracted keywords and sends the resulting cipher texts to server and keyword to the receiver, when the receiver wants to retrieve the file containing a specific keyword, he delegates a keyword search to trapdoor of server. Server finds the encrypted files containing the queried keywords without knowing the original file or keyword and returns the corresponding encrypted file to receiver and receiver decrypts file. The keyword searchable ciphertexts form hidden star-like structure as shown in fig. 3.

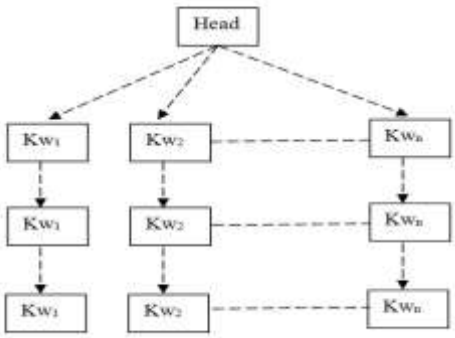


Fig. 3 Hidden star-like structures

Here the dashed arrows denote the hidden relations. Kw denotes the searchable Cipher text keyword. All cipher texts have same keywords that form chain by correlated hidden relation also hidden relation exists from public head to first cipher texts of each chain. With keyword search trapdoor and head, the server checks the first matching ciphertexts through the corresponding relation from head. By carrying this, all matching ciphertexts can be found. Thus search time depends on the actual number of ciphertexts containing the queried keyword rather than total number of all ciphertexts in cloud database.

2) FKSCH algorithm

(FKSCH): It consists of five algorithms:

- System Setup(1a, S): Take a security parameter and keyword space i.e. 1a and S respectively, and produce output as a pair of master public and secret keys (PuK, StK), where PuK includes the keyword space S and ciphertexts space Cs.
- Structured Initialization (PuK): Take PuK as input, and probabilistically initialization a hidden structure by outputting its private & public parts (Pr, Pb).
- Structured Encryption(PuK, S, Pr): Take PuK, a keyword V∈S and a hidden structure's private part Pr as input, and probabilistically output a eyword-searchable ciphertext Ct of keyword V with the hidden structure, and update Pr.
- Trapdoor(StK,V): Take StK and a keyword V∈S as input, and output keyword search trapdoor D of V.
- Structured Search (PuK, Pb, CtS, D): Take PuK , a hidden structure public part Pb, all keyword-searchable ciphertexts CtS and a keyword search trapdoor D of keyword V as input, disclose partial relations to guide finding out the ciphertexts containing keyword V with hidden structure.

3) A Simple FKSCH scheme from Scratch

Let $\beta \overset{\$}{\leftarrow} \phi$ depicts an element β randomly sampled from ϕ. M & M_1 denote two multiplicative Group of prime order ϕ. Let m be a generator of M. A bilinear Map ê:MxM→M_1 is an efficiently computable & non-degenerate function with the bi-linearity property $\hat{e}(m^c,m^d)= \hat{e}(m,m)^{cd}$, where (c,d) $\overset{\$}{\leftarrow} R_p^*$ and ê(m,m) is generator fo M_1.

Let BGen (1^a) be an efficient bilinear map generator which takes Security parameter 1^a as input and probabilistically outputs(p,M,M_1,m, ê). Let keyword space S=$\{0,1\}^*$.

A simple SPCHS scheme secure in the Random Oracle model is constructed as follows.

- System Setup(1^a,S): 1a and keyword space S is taken as input. Compute (p,M,M_1,m, ê)= BGen(1a), pick x $\overset{\$}{\leftarrow} R_p^*$ Set Q=mx, choose cryptographic hash function H:S→M, set the ciphertext space Cs $\sqsubseteq$ M_1*M*M_1 and finally outputs the master public key PuK=(p,M,M_1,m, ê,Q,H,S,Cs) and master secret key StK=x.

- Structure Initialization (Puk): Puk is taken as input , choose v$\xleftarrow{\$} R_p^*$ and initialize a hidden structure by outputting a pair of private-and-public parts (Pr=(v), Pb=mv). Note that Pr here is a variable list formed as (v, {(S,Qt[v,V])|(V∈ S)}), which is initialized as (v).
- Structured Encryption (PuK,V,Pr) : Take PuK, a keyword V∈S as input , a hidden structure private part Pr, pick z$\xleftarrow{\$} R_p^*$ and do the following steps:

1. Search (V,Qt[v,V]) for V in Pr

2. if it is not found, insert (V,Qt[v,V] $\xleftarrow{\$}$$M_1$) to Pr and output the keyword searchable ciphertext C_t=(ê(Q,H(V))^v, m^z, ê(Q,H(V))^z.Qt[v,V]);

3. Otherwise, pick B$\xleftarrow{\$}$$M_1$, set C_t= (Qt[v,V], m^z, ê(Q,H(V))^z. B), update Qt[v,V]=B, and output the keyword searchable ciphertext C_t.

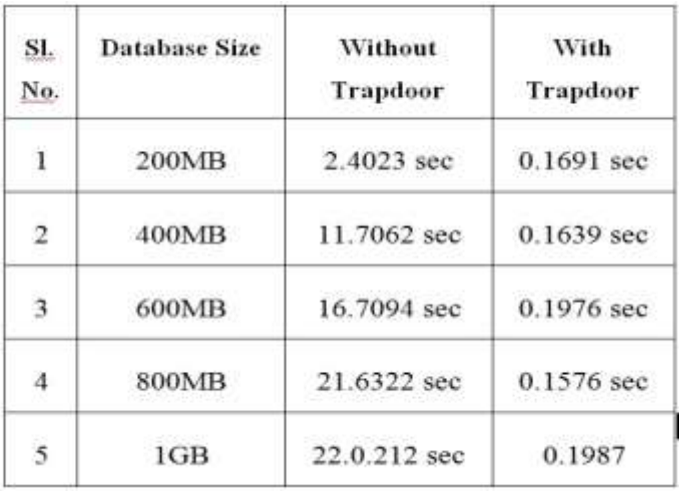
- Trapdoor(StK,V): take as input Stk and a keyword V∈ S, and output a keyword search trapdoor D=H(V)x of keyword V.
- Structured Search(PuK,Pb,CtS,D): PuK is taken as input, Pb as a hidden structure public part, CtS as all keyword searchable ciphertext and D as a keyword trapdoor of Keyword V, set CtS'= Ψ and do the following steps:
  1) Compute Qt'= ê(Pb, D);
  2) Seek a ciphertext CtS[i] having Cts[i,1] =Qt'; if it exists, add CtS[i] into CtS'.
  3) If no matching ciphertext is found, output CtS'.
  4) Compute Qt'= ê(CtS[i,2], D)-1. CtS[i,3] and go to step 2.

V. RESULTS

In previous section, we discussed the design issues to implement the idea of FKSCH. The models with trapdoor and without trapdoor are discussed elaborately. These models are implemented, executed and obtained results are tabulated in table 1. The corresponding graph is shown in fig. 4. The time taken with trapdoor algorithm takes less time for searching the file compared to without trapdoor algorithm as more and more the size of database grows.

TABLE 1

TIME TAKEN WITH TRAPDOOR AND WITHOUT TRAP DOOR



Sl. No.	Database Size	Without Trapdoor	With Trapdoor
1	200MB	2.4023 sec	0.1691 sec
2	400MB	11.7062 sec	0.1639 sec
3	600MB	16.7094 sec	0.1976 sec
4	800MB	21.6322 sec	0.1576 sec
5	1GB	22.0.212 sec	0.1987

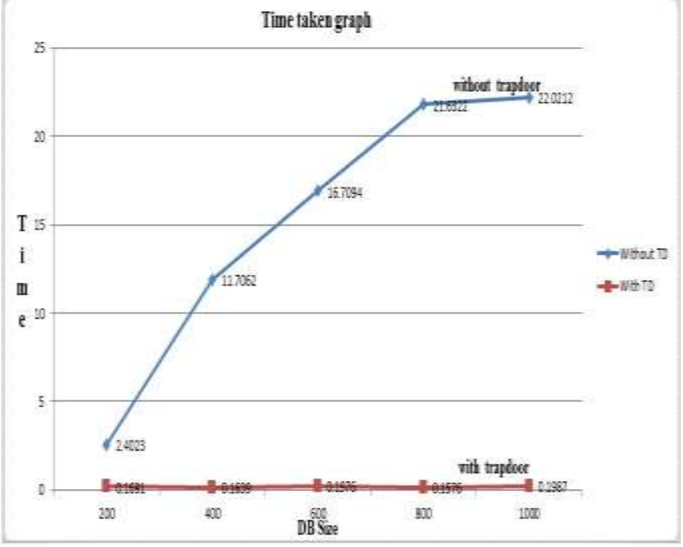


Fig. 4 Time taken with trapdoor and without trap door

VI. CONCLUSIONS

This paper explores the search technique and semantic security in cloud. We implemented the concept of FKSCH with trapdoor against without trapdoor algorithm. FKSCH allows keyword-search to be generated with a hidden structure. If the given keyword found in a trapdoor, the search algorithm of FKSCH can disclose part of this hidden structure for ushering towards the ciphertexts of the requested keyword. Semantic security of FKSCH holds the privacy of the keywords and the invisibility of the hidden structures. We implemented an FKSCH scheme from scratch with semantic security in the RO model and compared and proved better with other existing models. In spite of better performance our model should also have achieved the search completeness by letting ring structure.

Authors Information

Srinath K S B.E., M.Tech., MISTE., was born in Bangalore, India. He completed his Bachelor Degree in Engineering with specialization in Information Science and Masters Degree in Technology with specialization in Network and Internet Engineering from Visvesvaraya Technological University, Belgaum, Karanataka. Presently he is working as Assistant professor, Department of Computer Science & Engineering, Sambhram Institute of Technology, Bangalore, India. His areas of interest are Web Service, Web Security, Network security, cryptography and cloud computing.

GovindRaj J, pursuing BE, in Computer Science & Engineering from Sambhram Institute of Technology, Bangalore India. His area of interest are Data structures, formal language and automata theory, operation research.

Kirti Kumari, pursuing BE, in Computer Science & Engineering from Sambhram Institute of Technology, Bangalore India. Her area of interest are Data structures, formal language and automata theory, operation research.

Poojitha. M, pursuing BE, in Computer Science & Engineering from Sambhram Institute of Technology, Bangalore India. Her area of interest are Data structures, Database management system, formal language and automata theory, operation research.

Pintu Kumar, pursuing BE, in Computer Science & Engineering from Sambhram Institute of Technology, Bangalore India. His area of interest are Data structures, formal language and automata theory, operation research.

References

[1] Boneh D., Crescenzo G. D., Ostrovsky R., Persiano G.: Public Key Encryption with Keyword Search. In: Cachin C., Camenisch J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506-522. Springer, Heidelberg (2004).

[2] Bellare M., Boldyreva A., O'Neill A.: Deterministic and Efficiently Searchable Encryption. In: Menezes A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 535-552. Springer, Heidelberg (2007)

[3] Boneh D., Boyen X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin C., Camenisch J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223-238. Springer, Heidelberg (2004)

[4] Boyen X., Waters B. R.: Anonymous Hierarchical Identity-Based Encryption (Without Random Oracles). In: Dwork C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290-307. Springer, Heidelberg (2006)

[5] Huang H., Yang B., Zhu S., Xiao G. (2008) Generalized ElGamal Public Key Cryptosystem Based on a New Diffie-Hellman Problem. In: Baek J., Bao F., Chen K., Lai X. (eds) Provable Security. ProvSec 2008, vol 5324. Springer, Berlin, Heidelberg

[6] Li J., Lin Y., Wen M., Gu C., Yin B. (2015) Secure and Verifiable Multi-owner Ranked-Keyword Search in Cloud Computing. In: Xu K., Zhu H. (eds), vol 9204. Springer, Cham

[7] Ateniese G., Gasti P.: Universally Anonymous IBE Based on the Quadratic Residuosity Assumption. In: Fischlin M. (ed.) CT-RSA 2009. LNCS, vol. 5473, pp. 32-47. Springer, Heidelberg (2009)

[8] Ducas L.: Anonymity from Asymmetry: New Constructions for Anonymous HIBE. In: Pieprzyk J. (ed.) CT-RSA 2010. LNCS, vol. 5985, pp. 148-164. Springer, Heidelberg (2010)

[9] Abdalla M., Catalano D., Fiore D.: Verifiable Random Functions: Relations to Identity-Based Key Encapsulation and New Constructions. Journal of Cryptology, 27(3), pp. 544-593 (2013)

[10] Freire E.S.V., Hofheinz D., Paterson K.G., Striecks C.: Programmable Hash Functions in the Multilinear Setting. In: Canetti R., Garay J.A. (eds.) Advances in Cryptology – CRYPTO 2013. LNCS, vol. 8042, pp. 513-530. Springer, Heidelberg (2013)