ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 12, July 2017

Two Independent Server Public Key Encryption and Keyword Search for Secure Cloud Access

Swetha N[1], Prof. Sreelatha P K [2]

P G Student, Dept. of CSE, SVIT College, Rajanukunte, Bengaluru, India[1]

Asst. Prof, Dept. of CSE, SVIT College, Rajanukunte, Bengaluru, India [2]

ABSTRACT: In this work, we investigate the security of a well-known cryptographic primitive, namely Public Key Encryption with Keyword Search 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 launched by the malicious server. To address this security vulnerability, we propose a new PEKS framework named Dual-Server Public Key Encryption with Keyword Search. 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). To illustrate the feasibility of we provide an efficient instantiation of the general framework from a DDH-based LH-SPHF and show that it can achieve the strong security against inside KGA.

KEYWORDS: Keyword search, secure cloud storage, encryption, inside keyword guessing attack, smooth projective hash function.

I. INTRODUCTION

Cloud storage outsourcing has become a popular application for enterprises and organizations to reduce the burden of maintaining big data in recent years. However, in reality, end users may not entirely trust the cloud storage servers and may prefer to encrypt their data before uploading them to the cloud server in order to protect the data privacy. This usually makes the data utilization more difficult than the traditional storage where data is kept in the absence of encryption. One of the typical solutions is the searchable encryption which allows the user to retrieve the encrypted documents that contain the user-specified keywords, where given the keyword trapdoor, the server can find the data required by the user without decryption.

A proposed keyword search on ciphertext, known as Searchable Symmetric Encryption and afterwards several SSE schemes [2], [3] were designed for improvements. Although SSE schemes enjoy high efficiency, they suffer from complicated secret key distribution. Precisely, users have to securely share secret keys which are used for data encryption. Otherwise they are not able to share the encrypted data outsourced to the cloud. To resolve this problem, Boneh etal. [4] introduced a more flexible primitive, namely Public Key Encryption with Keyword Search (PEKS) that enables a user to search encrypted data in the asymmetric encryption setting. In a PEKS system, using the receiver's public key, the sender attaches some encrypted keywords 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.

II. MOTIVATION OF THIS WORK

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. The reason leading to such 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

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 12, July 2017

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. Such a guessing attack has also been considered in many password-based systems. However, the attack can be launched more efficiently against PEKS schemes since the keyword space is roughly the same as a normal dictionary which has a much smaller size than a password dictionary (e.g., all the words containing 6 alphanumeric characters). It is worth noting that in SSE schemes, only secret key holders can generate the keyword ciphertext and hence the adversarial server is not able to launch the inside KGA. As the keyword always indicates the privacy of the user data, it is therefore of practical importance to overcome this security threat for secure searchable encrypted data outsourcing.

III. DIFFERENCES BETWEEN THIS WORK AND ITS PRELIMINARY VERSION

Portions of the work presented in this paper have previously appeared as an extended abstract [1]. Compared to [1], we have revised and enriched the work substantially in the following aspects. First, in the preliminary work [1] where our generic DS-PEKS construction was presented, we showed neither a concrete construction of the linear and homomorphic SPHF nor a practical instantiation of the DS-PEKS framework. To fill this gap and illustrate the feasibility of the framework, in this paper we first show that a linear and homomorphic language LDH can be derived from the Diffie-Hellman assumption and then construct a concrete linear and homomorphic SPHF, referred to as SPHFDH, from LDH. We provide a formal proof that SPHFDH is correct, smooth and pseudo-random and hence can be used for the instantiation of our generic construction. We then present a concrete DS-PEKS scheme from SPHFDH. To analyze its performance, we first give a comparison between existing schemes and our scheme and then evaluate its performance in experiments. We also revised the preliminary version [1] to enhance the presentation and readability. In the related work part, compared to the preliminary version, we add more literatures and give a clearer classification of the existing schemes based on their security. We present the security models of DS-PESK as experiments to make them more readable. Moreover, to make the concepts of SPHF and our newly defined variant clearer, we add to highlight their key properties.

IV. A NEW FRAMEWORK FOR PEKS

In this section, we formally define the Dual-Server Public Key Encryption with Keyword Search (DS-PEKS) and its security model.

4.1 Definition of DS-PEKS

A DS-PEKS scheme mainly consists of (KeyGen, DS-PEKS, DS-Trapdoor; FrontTest; BackTest). To be more precise, the KeyGen algorithm generates the public/private key pairs of the front and back servers instead of that of the receiver. Moreover, the trapdoor generation algorithm DS-Trapdoor defined here is public while in the traditional PEKS definition [4], [8], the algorithm Trapdoor takes as input the receiver's private key. Such a difference is due to the different structures used by the two systems. In the traditional PEKS, since there is only one server, if the trapdoor generation algorithm is public, then the server can launch a guessing attack against a keyword ciphertext to recover the encrypted keyword. As a result, it is impossible to achieve the semantic security as defined in [4], [8]. However, as we will show later, under the DS-PEKS framework, we can still achieve semantic security when the trapdoor generation algorithm is public. Another difference between the traditional PEKS and our proposed DS-PEKS is that the test algorithm is divided into two algorithms, FrontTest and BackTest run by two independent servers. This is essential for achieving security against the inside keyword guessing attack. In the DS-PEKS system, upon receiving a 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. 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.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 12, July 2017

4.2 Smooth projective hash functions

A central element of our construction for dual-server public key encryption with keyword search is smooth projective hash function (SPHF), a notion introduced by Cramer and Shoup [7]. We start with the original definition of an SPHF.

4.2.1 Original Definition of SPHFs

As illustrated in Fig. 1, an SPHF is defined based on a domain X and an NP language L, where L contains a subset of the elements of the domain X, i.e., L _ X. Formally, an SPHF system over a language L _ X, onto a set Y, is defined by the following five algorithms (SPHFSetup; HashKG, ProjKG; Hash; ProjHash):

- SPHFSetup(1): generates the global parameters param and the description of an NP language instance L;
- HashKG(L; param): generates a hashing key hk for L;
- ProjKG(hk; (L; param)): derives the projection key hp from the hashing key hk;
- Hash(hk; (L; param);W): outputs the hash value hv 2 Y for the word W from the hashing key hk;

ProjHash(hp; (L; param);W;w): outputs the hash value hv0 2 Y for the word W from the projection key hp and the witness w for the fact that W 2 L.

In summary, an SPHF has the property that the projection key uniquely determines the hash value of any word in the language L but gives almost no information about the hash value for any point in X n L. In this paper, we require another important property of smooth projective hash functions that was introduced in [5].Precisely, we require the SPHF to be pseudo-random. That is, if a word W 2 L, then without the corresponding witness w, the distribution of the hash output is computationally indistinguishable from a uniform distribution in the view of any polynomial-time adversary.

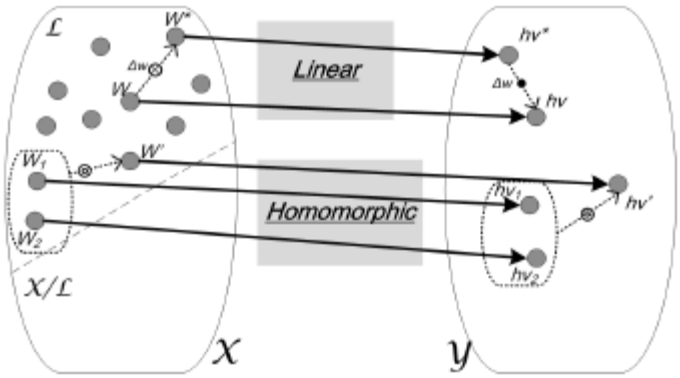

Fig1. smooth projective hash functions

V. EXPERIMENT RESULTS

The following experiments are based on coding language C on Linux system (more precise, 2.6.35-22-generic version) with an Intel(R) Core(TM) 2 Duo CPU of 3.33 GHZ and 2.00-GB RAM. As shown in Fig. 2, our scheme is the most efficient in terms of PEKS computation. It is because that our scheme does not include pairing computation. Particularly, the scheme [6] requires the most computation cost due to 2 pairing computation per PEKS generation. As for the trapdoor generation indicated in Figure 3, as all the existing schemes do not involve pairing computation, the computation cost is much lower than that of PEKS generation. As illustrated in Fig. 4, the scheme [6] cost the most time due to an additional pairing computation in the exact testing stage. One should note that this additional pairing computation is done on the user side instead of the server. Therefore, it could be the computation burden for users who may use a light device for searching data. 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.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 12, July 2017

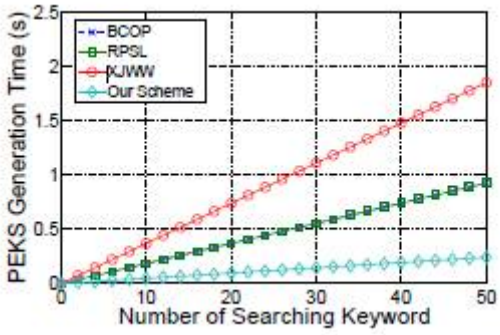

Fig2 Computation Cost of PEKS Generation in Different Schemes

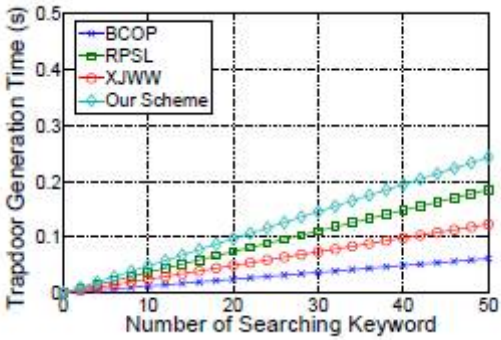

Fig. 3 Computation Cost of Trapdoor Generation in Different Schemes

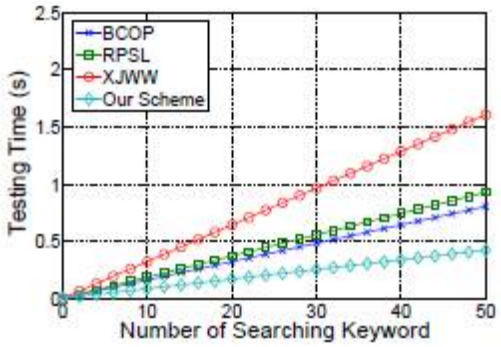

Fig4 Computation Cost of Testing in Different Schemes

VI. CONCLUSION

In this paper, we proposed a new framework, named Dual- Server Public Key Encryption with Keyword Search (DSPEKS), that can prevent the inside keyword guessing attack which is an inherent vulnerability of the traditional PEKS framework. We also introduced a new Smooth Projective Hash Function (SPHF) and used it to construct a generic DSPEKS scheme. An efficient instantiation of the new SPHF based on the Diffie-Hellman problem is also presented in which gives an efficient DS-PEKS scheme without pairings.

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 Information Security and Privacy - 20th Australasian Conference, ACISP, 2015,
[2] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order-preserving encryption for numeric data," in Proceedings of the ACM SIGMOD International Conference on Management of Data, 2004, pp. 563–574.
[3] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, 2006, pp. 79–88.
[4] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in EUROCRYPT, 2004, pp. 506–522.
[5] R. Gennaro and Y. Lindell, "A framework for password-based authenticated key exchange," in EUROCRYPT, 2003, pp. 524–543.
[6] 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 Trans. Computers, vol. 62, no. 11, pp. 2266–
2277, 2013.
[7] R. Cramer and V. Shoup, "Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption," in EUROCRYPT, 2002, pp. 45–64.
[8] J . Baek, R. Safavi-Naini, and W. Susilo, "Public key encryption with keyword search revisited," in Computational Science and Its Applications - ICCSA, 2008, pp. 1249–1259.