SERVER-AIDED PUBLIC KEY ENCRYPTION WITH KEYWORD SEARCH

N.Gireesh[1], Dr.M.Sreedevi[2]
[2]M.C.A, M.Phil., PhD

Abstract—**Public Key Encryption with Keyword Search (PEKS) is a well-known cryptographic primitive for secure searchable data encryption in cloud storage. Unfortunately, it is inherently subject to the (inside) off-line keyword guessing attack (KGA), which is against the data privacy of users. Existing countermeasures for dealing with this security issue mainly suffer from low efficiency and are impractical for real applications. In this work, we provide a practical and applicable treatment on this security vulnerability by formalizing a new PEKS system named Server-Aided Public Key Encryption with Keyword Search (SA-PEKS). In SA-PEKS, to generate the keyword cipher text/trapdoor, the user needs to query a semi-trusted third party called Keyword Server (KS) by running an authentication protocol and hence security against the off-line KGA can be obtained. We then introduce a universal transformation from any PEKS scheme to a secure SA-PEKS scheme using the deterministic blind signature. To illustrate its feasibility, we present the first instantiation of SA-PEKS scheme by utilizing the FDH-RSA signature and the PEKS scheme proposed by Boneh et al. in Euro crypt 2004. Finally, we describe how to securely implement the client-KS protocol with a rate-limiting mechanism against on-line KGA and evaluate the performance of our solutions in experiments.**

Index Terms—Public key encryption with keyword search, server-aided, off-line keyword guessing attack.

I. INTRODUCTION

Cloud storage outsourcing is of increasing interest in recent years for enterprises and organizations to reduce the burden of maintaining big data. In reality, end users may prefer to encrypt their outsourced data for privacy protection as they may not entirely trust the cloud storage server. This makes deployment of traditional data utilization service, such as plaintext keyword search over textual data or query over database, a difficult task. One of the typical solutions is the searchable encryption which allows the user to search and retrieve the encrypted data, and meanwhile preserve the data privacy.

Unfortunately, despite being free from secret key distribution, PEKS schemes suffer from an inherent security problem regarding the keyword privacy, namely (inside) off-line Keyword Guessing Attack (KGA). 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 cipher text. 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. As the keyword always could leak some sensitive information of the user data, it is therefore of practical importance to overcome this security threat for secure and searchable encrypted data outsourcing.

A. Existing System:

We aim at designing a more practical treatment to address this security issue. Moreover, we are interested in building a system that works transparently with any existing PEKS system. That is, the system will be backward-compatible and make no modification on the implementation details of the underlying PEKS system.

We formalize a new PEKS system named Server-Aided Public Key Encryption with Keyword Search (SA-PEKS) to address the security vulnerability against (inside) offline KGA in existing PEKS systems.

We remark that this result is independent of the underlying PEKS scheme, as our solution works transparently with any existing PEKS system.

The feasibility of such a solution is due to that our proposed solution can work transparently with any existing PEKS system and hence the above universal transformation is also applicable for the SCF-PEKS.

B. Proposed System:

The notion of Public-key Encryption with Fuzzy Keyword Search (PEFKS) where each keyword corresponds to an exact trapdoor and a fuzzy trapdoor.

The server is only provided with the fuzzy trapdoor and thus can no longer learn the exact keyword.

Our proposed solution can work transparently with any existing PEKS system and hence is much more applicable in practice. Secondly, we present a generic construction of SA-PEKS scheme with formal security analysis. Precisely, we propose a universal transformation from any PEKS scheme to an SA-PEKS scheme by utilizing a deterministic blind signature.

The feasibility of the proposed generic transformation, an instantiation of the SA-PEKS scheme is presented in this paper. We instantiate the scheme from the FDH-RSA blind signature and the PEKS scheme.

II. IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most easiest stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to him/her work very easy.

A) Modules:

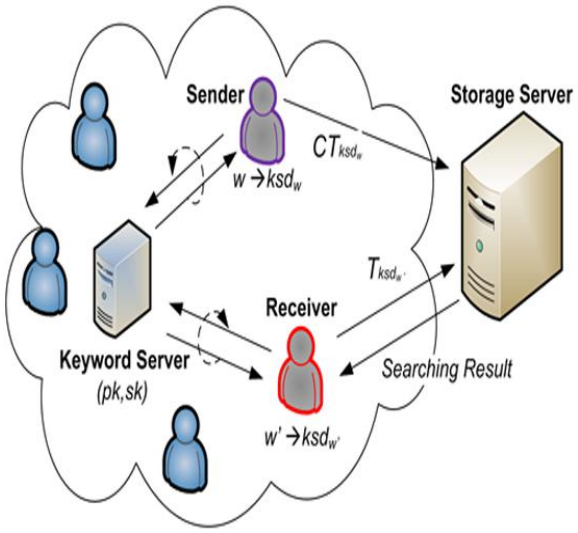
1. Server-Aided PEKS
2. PEKS-To-SA-PEKS Transformation
3. Instantiations of SA-PEKS

1. Server-Aided PEKS:

SA-PEKS is motivated by the observation that the offline KGA can be dealt with by employing a semi-trusted third party, namely Keyword Server (KS) which is separated from the Storage Server (SS).

Roughly speaking, in an SA-PEKS system, the KS owns the public/ secret key pair (pk; sk). Users authenticate themselves to the KS and are provisioned with per-user credentials. Different from the PEKS framework where the PEKS cipher text and the trapdoor are derived from the original keyword directly, the user needs to interact with the KS in an authenticated way to obtain the pre-processed keyword, namely KS-derived keyword, before the generation of the PEKS cipher text and the trapdoor.

Adversarial Storage Server (SS) is a new notion, namely Semantic-Security against Chosen Keyword Guessing Attack (SS-CKGA) for the SA-PEKS. Similar to the notion of SS-CKA in PEKS, SS-CKGA guarantees that the PEKS cipher text in the SA-PEKS does not reveal any information about the underlying keyword. The difference between the SS-CKGA and SS-CKA is that the adversary against the SA-PEKS is allowed to obtain the matching trapdoor of the challenge PEKS cipher text.



System model of Server-Aided PEKS

2. PEKS-To-SA-PEKS Transformation:

The security of blind signature is twofold: unforgeability and blindness. We say a blind scheme is one-more unforgeable if any polynomial time adversary that queries the signing oracle with q_s distinct messages can only forge q_s+1 valid message/signature pairs with negligible probability. Another notion, namely blindness, requires that the signer cannot tell apart the message it is signing. To be more precise, the blindness condition says that it should be infeasible for a malicious signer to decide which of the two messages has been signed first in two executions with an honest user.

One may concern that the curious KS could also lunch the off-line KGA by intercepting the transferred trapdoor from the communication channel between the SS and the receiver. To achieve the stronger security against such a curious KS, we could follow the idea of secure channel free PEKS (SCFPEKS). The feasibility of such a solution is due to that our proposed solution can work transparently

with any existing PEKS system and hence the above universal transformation is also applicable for the SCF-PEKS.

3. Instantiations of SA-PEKS:

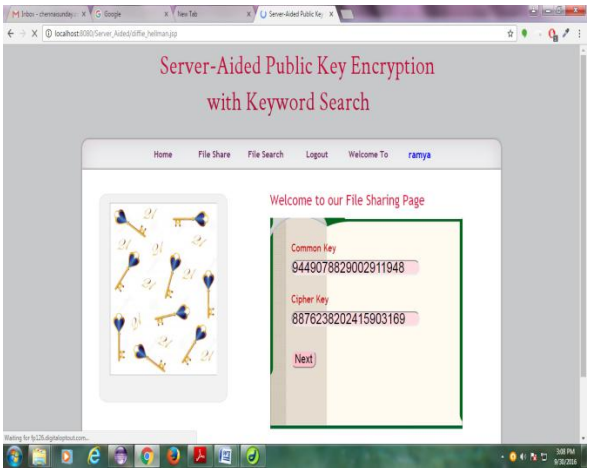
The security of the resulting SA-PEKS scheme can be easily obtained based on as the FDH-RSA is one-more-unforgivable and of blindness.

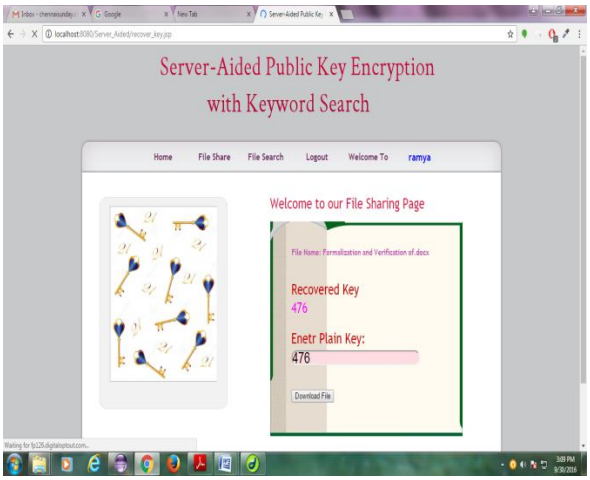
A protocol for client-KS interaction and the rate-limiting strategies which limit client queries to slow down on-line keyword guessing attack. Our design goal is to give a low-latency protocol to avoid performance degrading. The proposed protocol relies on a CA providing the KS and each client with a unique verifiable TLS certificate the execution of protocol consists of the Mutual Authentication (MA) phase and the Query-Response (QR) phase, of which the first one is over HTTP while the later one is over UDP.

III. ALGORITHMS

Diffie–Hellman key exchange (D–H) is a specific method of securely exchanging cryptographic keys over a public channel and was one of the first public-key protocols as originally conceptualized by Ralph Merkle and named after Whitfield Diffie and Martin Hellman. D–H is one of the earliest practical examples of public key exchange implemented within the field of cryptography.

Traditionally, secure encrypted communication between two parties required that they first exchange keys by some secure physical channel, such as paper key lists transported by a trusted courier. The Diffie–Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher





IV. INPUT AND OUTPUT DESIGNS

A. Input Design:

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy.

Input Design considered the following things:

What data should be given as input?

How the data should be arranged or coded?

The dialog to guide the operating personnel in providing input.
Methods for preparing input validations and steps to follow when error occur.

B. Objectives:
Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user
Will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

C. Output Design:
A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.
Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
Select methods for presenting information.
Create document, report, or other formats that contain information produced by the system.
The output form of an information system should accomplish one or more of the following objectives.
Convey information about past activities, current status or projections of theFuture.Signal important events, opportunities, problems, or warnings. Trigger an action. Confirm an action.

V. SYSTEM STUDY

A) Feasibility Study:
The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.
Three key considerations involved in the feasibility analysis are

1. Economical Feasibility
2. Technical Feasibility
3. Social Feasibility

1. Economical Feasibility:
This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2. Technical Feasibility:
This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3. Social Feasibility:
The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the

system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

VI. LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy n company strength. Once these things r satisfied, ten next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration r taken into account for developing the proposed system.

Public Key Encryption with Keyword Search (PEKS) is a well-known cryptographic primitive for secure searchable data encryption in cloud storage. Unfortunately, it is inherently subject to the (inside) off-line keyword guessing attack (KGA), which is against the data privacy of users. Existing countermeasures for dealing with this security issue mainly suffer from low efficiency and are impractical for real applications. In this work, we provide a practical and applicable treatment on this security vulnerability by formalizing a new PEKS system named Server-Aided Public Key Encryption with Keyword Search (SA-PEKS).

In SA-PEKS, to generate the keyword cipher text/trapdoor, the user needs to query a semi-trusted third party called Keyword Server (KS) by running an authentication protocol and hence security against the off-line KGA can be obtained. We then introduce a universal transformation from any PEKS scheme to a secure SA-PEKS scheme using the deterministic blind signature. To illustrate its feasibility, we present the first instantiation of SA-PEKS scheme by utilizing the FDH-RSA signature and the PEKS scheme proposed by Boneh et al. in Euro crypt 2004. Finally, we describe how to securely implement the client-KS protocol with a rate-limiting mechanism against on-line KGA and evaluate the performance of our solutions in experiments.

We aim at designing a more practical treatment to address this security issue. Moreover, we are interested in building a system that works transparently with any existing PEKS system. That is, the system will be backward-compatible and make no modification on the implementation details of the underlying PEKS system.

We formalize a new PEKS system named Server-Aided Public Key Encryption with Keyword Search (SA-PEKS) to address the security vulnerability against (inside) offline KGA in existing PEKS systems.

We remark that this result is independent of the underlying PEKS scheme, as our solution works transparently with any existing PEKS system. The feasibility of such a solution is due to that our proposed solution can work transparently with any existing PEKS system and hence the above universal transformation is also applicable for the SCF-PEKS.

The notion of Public-key Encryption with Fuzzy Keyword Search (PEFKS) where each keyword corresponds to an exact trapdoor and a fuzzy trapdoor. The server is only provided with the fuzzy trapdoor and thus can no longer learn the exact keyword.

Our proposed solution can work transparently with any existing PEKS system and hence is much more applicable in practice. Secondly, we present a generic construction of SA-PEKS scheme with formal security analysis. Precisely, we propose a universal transformation from any PEKS scheme to an SA-PEKS scheme by utilizing a deterministic blind signature.

The feasibility of the proposed generic transformation, an instantiation of the SA-PEKS scheme is presented in this paper. We instantiate the scheme from the FDH-RSA blind signature and the PEKS scheme.

SA-PEKS is motivated by the observation that the offline KGA can be dealt with by employing a semi-trusted third party, namely Keyword Server (KS) which is separated from the Storage Server (SS).

Roughly speaking, in an SA-PEKS system, the KS owns the public/ secret key pair (pk; sk). Users authenticate themselves to the KS and are provisioned with per-user credentials. Different from the PEKS framework where the PEKS cipher text and the trapdoor are derived from the original keyword directly, the user needs to interact with the KS in an authenticated way to obtain the pre-processed keyword, namely KS-derived keyword, before the generation of the PEKS cipher text and the trapdoor.

Adversarial Storage Server (SS) is a new notion, namely Semantic-Security against Chosen Keyword Guessing Attack (SS-CKGA) for the SA-PEKS. Similar to the notion of SS-CKA in PEKS, SS-CKGA guarantees that the PEKS cipher text in the SA-PEKS does not reveal any information about the underlying keyword. The difference between the SS-CKGA and SS-CKA is that the adversary against the SA-PEKS is allowed to obtain the matching trapdoor of the challenge PEKS cipher text.

The security of blind signature is twofold: unforgeability and blindness. We say a blind scheme is one-more unforgeable if any polynomial time adversary that queries the signing oracle with q_s distinct messages can only forge q_s+1 valid message/signature pairs with negligible probability. Another notion, namely blindness, requires that the signer cannot tell apart the message it is signing. To be more precise, the blindness condition says that it should be infeasible for a malicious signer to decide which of the two messages has been signed first in two executions with an honest user.

One may concern that the curious KS could also lunch the off-line KGA by intercepting the transferred trapdoor from the communication channel between the SS and the receiver. To achieve the stronger security against such a curious KS, we could follow the idea of secure channel free PEKS (SCFPEKS). The feasibility of such a solution is due to that our proposed solution can work transparently with any existing PEKS system and hence the above universal transformation is also applicable for the SCF-PEKS.

The security of the resulting SA-PEKS scheme can be easily obtained based on as the FDH-RSA is one-more-unforgivable and of blindness.

A protocol for client-KS interaction and the rate-limiting strategies which limit client queries to slow down on-line keyword guessing attack. Our design goal is to give a low-latency protocol to avoid performance degrading. The proposed protocol relies on a CA providing the KS and each client with a unique verifiable TLS certificate the execution of protocol consists of the Mutual Authentication (MA) phase and the Query-Response (QR) phase, of which the first one is over HTTP while the later one is over UDP.

VII. CONCLUSION

We provided a practical and applicable treatment on (inside) off-line KGA by formalizing a new PEKS system, namely Server-Aided Public Key Encryption with Keyword Search (SA-PEKS). We introduced a universal transformation from any PEKS scheme to a secure SAPEKS scheme, also with the first instantiation of SA-PEKS scheme and showed how to securely implement the client- KS protocol with a rate-limiting mechanism against on-line KGA. The experimental results showed that our proposed scheme achieves much better efficiency while providing resistance against both off-line and on-line KGAs.

VIII. ACKNOWLEDGEMENT

We thank the anonymous reviewers for their insightful feedbacks on this work. The work of Yongjun Wangis supported by the National Natural Science Foundation of China (Grant No. 61472439).

REFERENCES

[1] D. X. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in IEEE Symposium on Security and Privacy, 2000, pp,44–55.

[2] R.CurtmolaJ.A.Garay, S.Kamara, and R.Ostrovsky, “Searchable symmetric Encryption: improved definitions and efficient constructions,” in ACM CCS, 2006, pp. 79–88.

[3] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with

keyword search," in EUROCRYPT, 2004, pp. 506–522.

[4] X. Yi, E. Bertino, J. Vaidya, and C. Xing, "Private searching on streaming data based on keyword frequency," IEEE Transactions on Dependable and Secure Computing, vol. 11, no. 2, pp. 155–167, 2014.

[5] 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.

[6] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "A new general framework for secure public key encryption with keyword search," in ACISP, 2015, pp. 59–76.

[7] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "Dual-server public-key encryption with keyword search for secure cloud storage," Public key encryption with keyword search, vol.11, no. 4, pp. 789–798, 2016.

[8] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, "Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions," in CRYPTO, 2005, pp. 205–222.

[9] D. Khader, "Public key encryption with keyword search based on k-resilient IBE," in Computational Science and Its Applications ICCSA, 2006, pp. 298–308.

[10] P. Xu, Q. Wu, W. Wang, W. Susilo, J. Domingo-Ferrer, and H. Jin, "Generating searchable public-key ciphertexts with hidden structures for fast keyword search," Public key encryption with keyword search, vol. 10, no. 9, pp. 1993–2006, 2015.