IARJSET
International Advanced Research Journal in Science, Engineering and Technology
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
NCRTET-17
National Conference on Recent Trends in Engineering & Technology
Government Polytechnic College Kasaragod, Kerala
Vol. 4, Special Issue 1

Searching On Encrypted Data With The Help Of A Keyword Server

DEEPU MATHEW[1], **HIMA ANNS ROY**[2]
M.Tech, Computer Science and Engineering, S.J.C.E.T, Palai, India [1]
Assistant Professor, Computer Science and Engineering, S.J.C.E.T, Palai,India [2]

Abstract:In this modern computational age every devices generating data. Processing and storing the mass amount of data become an issue. To solve this problems every one use cloud storage. Because of privacy user favor encrypted data for storing on the cloud. Searching performed on encrypted data is not like an ordinary searching. Public key encryption with keyword search is one of the traditional methods to perform the searching on encrypted data. But it is vulnerable to the offline keyword guessing attack. In this work, searching performed with the help of a keyword server which provides keywords for searching and avoids such kind of keyword guessing attacks. The keyword server acts like a trusted third person and share the keys of users with the server. Every time user uploads their data into the database, a randomly generated keyword is also stored along with that data. Public key encryption technique ECC (Elliptic Curve Cryptography) is used to perform encryption.

Keywords: KGA (Keyword Guessing Attack) Keyword Server , ECC (Elliptic Curve Cryptography)

I. INTRODUCTION

In this modern world everyone use computer for different purposes. Due to this reason huge amount of data are generated in a single day. Processing and storage of such data become a serious issue because of the limited storage capacity of commodity hardware. So in such situation the term cloud storage comes into action. Because of privacy user favour encrypted data for storing on the cloud. For encryption public key or private key encryption methods are used. Searching on encrypted data is not a simple task. In traditional method user simply place a keyword along with the encrypted data. Then the user who wishes to search that data is using the keyword saved during the time of uploading. Public key encryption with keyword search is one of the traditional methods to perform the searching on encrypted data. But it is vulnerable to the offline keyword guessing attack. Keyword guessing attack done by offline leads to the leakage of data and other security issues. In this Paper a new method is introduced to avoid such kind of offline keyword guessing attacks with the help of a keyword server. The keyword server is always online so offline attacks are not possible. And this system generates the keyword for searching is not related to the actual content of the data. In traditional methods the keywords are some words generated from the actual content so it leads to guessing of keywords.

II. RELATED WORK

Secure Channel-Free PEKS. Baek et al. [1]proposed a new PEKS scheme, which is referred to as secure channel-free PEKS (SCF-PEKS). Rhee et al. [2] later enhanced Baek et al.'s security model [1] for SCF-PEKS where the attacker is allowed to obtain the relationship between the non-challenge cipher texts and the trapdoor. They also presented an SCF-PEKS scheme secure under the enhanced security model in the random oracle model.

*Traditional PEKS.*Following Boneh et al.'s seminalwork [3], Abdalla et al. [4] formalized anonymous IBE (AIBE) and presented a generic construction of searchable encryption from AIBE. In order to construct a PEKS secure in the standard model, Khader [5] proposed a scheme based on the k-resilient IBE. In [10], an interesting primitive called searchable public-key cipher texts with hidden structures (SPCHS) was proposed for efficient keyword search without sacrificing semantic security of the encrypted keywords.

Against Outside KGA. Byun et al. [6] introduced theoffline 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. Inspired by the work of Byun et al. [6], Yau et al. [7] demonstrated that outside adversaries that capture the trapdoors sent in a public channel can reveal the encrypted keywords through off-line keywordguessing attacks and they also showed off-line keyword guessing attacks against the (SCF-)PEKS schemes in [1], [8]. The first PEKS scheme secure against outside keyword guessing attacks was proposed by Rhee et al. [9].

III.PROPOSED SYSTEM

A. An Overview

B. The Searching on encrypted data with the help of a keyword server is motivated by the observation thatthe 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. More specifically, given an original keyword w, the sender has to access the KS through authentication and run an interactive protocol with the KS. At the end of the protocol execution, the sender gets the corresponding KS-derived keyword.

C. Proposed Architecture

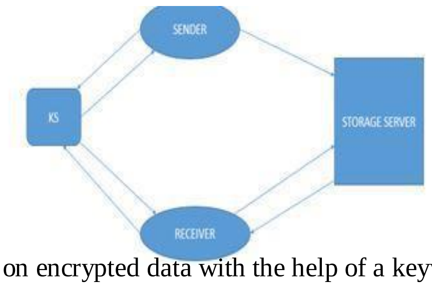


Fig. 1: Searching on encrypted data with the help of a keyword server.

Figure1 shows the architectural diagram of the system. The system contains 3 modules namely user, storage server and keyword server. User module deals with the activities of user. This module contains the functions such as uploading data, search the data that posted by the other user .Storage server is the person who manages the domain. This module provides the facilities for the admin for managing users, providing key pairs etc… Keyword server is a person who provide the keywords to the user who wants to search on encrypted data.

D. Work Flow

Step1:The KS (keyword server owns the public/secret key pair.
Step2: The sender sends original keyword to server
Step3: The sender gets the corresponding KS-derived keyword of w as ksdw
Step4:The sender then generates the PEKS ciphertext by regarding the KS-derived keyword ksdw as the final keyword.
Step5:The receiver interact with KS to obtain ksdw'(input is the keyword w') And then generates the corresponding trapdoor.

IV. IMPLEMENTATION

A. Functional Requirements

- Anybody can use the system but the requirement is a valid username and password.
- The system have a database which contains the information about the all users, admin etc…
- The system check the genunity of the user.

B. Non Functional Requirements

a) Hardware Requirements

Processor	:P4
RAM	:2GB
Hard Disc	:500GB
Monitor	:14'SVGA
CD Drive	: 52X

b) Software Requirements

Operating System	: Windows-XP
Front-end	: JAVA
Back-end	:My SQL
IDE	:Netbeans

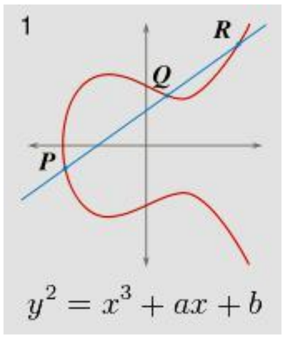
C. Methodology

a) Elliptic Curve Cryptography

All Elliptic Curve Cryptography (ECC) was discovered in 1985 by Victor Miller (IBM) and Neil Koblitz (University of Washington) as an alternative mechanism for implementing public-key cryptography.

$$y^2 = x^3 + ax + b$$

Few terms that will be used,
E -> Elliptic Curve
P -> Point on the curve
n -> Maximum limit (This should be a prime number)



<u>Key Generation</u>

Key generation is an important part where we have to generate both public key and private key. The sender will be encrypting the message with receiver's public key and the receiver will decrypt its private key.
Now, we have to select a number **'d'** within the range of **'n'**.
Using the following equation we can generate the public key
The equation of an elliptic curve is given as,
Q = d * P
d = The random number that we have selected withinthe range of (**1 to n-1**). **P** is the point on the curve. 'Q' is the public key and 'd' is the private key.

<u>Encryption</u>
Let 'm' be the message that we are sending. We have to represent this message on the curve. This have in-depth implementation details. All the advance research on ECC is done by a company called certicom. Consider *'m'* has the point *'M'* on the curve *'E'*. Randomly select 'k' from [1 – (n-1)].

Two cipher texts will be generated let it be **C1** and **C2**.

C1 = k*P
C2 = M + k*Q
C1 and C2 will be send.

<u>Decryption</u>
We have to get back the message 'm' that was send to us,

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

M = C2 – d * C1
M is the original message that we have send.

Proof
How does we get back the message,

M = C2 – d * C1

'M' can be represented as 'C2 – d * C1'

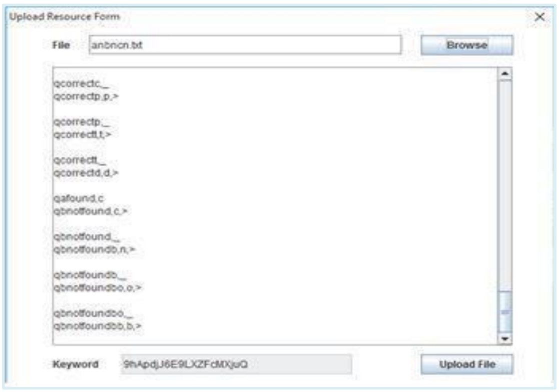
C2 – d * C1 = (M + k * Q) – d * (k *P) (C2 = M + k * Q and C1 = k * P)

= M + k * d * P – d * k *P (canceling out k *d * P)

= M (Original Message)

D. Bouncy Castle FIPS Java API

This API is used to implement cryptographic algorithms. The Bouncy Castle APIs (BC) divide into 3 groups: there is a light-weight API which provides direct access to cryptographic services, a JCA/JCE provider built on top of the light-weight API that provides access to services required to use the JCA/JCE, and another set of APIs which provide handling of protocols such as Cryptographic Message Syntax (CMS), OpenPGP, Time Stamp Protocol (TSP), Secure Mime (S/MIME), Certificate Management Protocol (CMP), as well as APIs forgenerating Certification Requests (CRMF, PKCS#10), X.509 certificates, PKCS#12 files and other protocol elements used in a variety of standards. The total code base, including porting code for different JVMs, is currently sitting at 499,000 lines of Java.In terms of an overall design the APIs have been put together in a manner that allows the different classes defined to be used to create objects which can be assembled in a variety of ways, regardless of whether it makes sense.

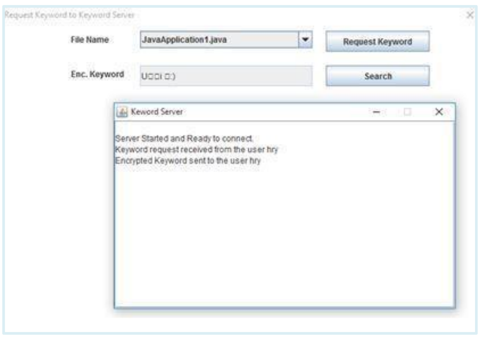


File: anbncn.txt

```
qcorrectc,_
qcorrectp,p,>

qcorrectp,_
qcorrectt,t,>

qcorrectt,_
qcorrectd,d,>

qafound,c
qbnotfound,c,>

qbnotfound,_
qbnotfoundb,n,>

qbnotfoundb,_
qbnotfoundbo,o,>

qbnotfoundbo,_
qbnotfoundbb,b,>
```

Keyword: 9hApdjJ6E9LXZFcMXjuQ

Fig.2. Generation of keyword.



File Name: JavaApplication1.java
Enc. Keyword: U□□i □:)

```
Server Started and Ready to connect.
Keyword request received from the user hry
Encrypted Keyword sent to the user hry
```

Fig.3. Keyword send to the user

RESULTS AND DISCUSSIONS

In this work, provided a practical and applicable treatment on (inside) off-line KGA by formalizing anew PEKS system, namely Searching on encrypted data with the help of a keyword server. The experimental results showed that our proposed scheme achieves much better efficiency while providing resistance against both off-line and on-line Keyword guessing attacks.

REFERENCES

[1] J. Baek, R. Safavi-Naini, and W. Susilo, "Public key encryption with keyword search revisited," in Computational Science and ItsApplications - ICCSA, 2008, pp. 1249–1259.
[2] H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee, "Improved searchable public key encryption with designated tester," inASIACCS,2009, pp. 376–379.
[3] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," inEUROCRYPT, 2004, pp.506–522.
[4] 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.
[5] D. Khader, "Public key encryption with keyword search based on k-resilient IBE," in Computational Science and Its Applications -ICCSA, 2006, pp. 298–308.
[6] J. W. Byun, H. S. Rhee, H. Park, and D. H. Lee, "Off-line keyword guessing attacks on recent keyword search schemes over encrypted data," in Secure Data Management, Third VLDB Workshop, SDM, 2006, pp. 75–83.
[7] W. Yau, S. Heng, and B. Goi, "Off-line keyword guessing attacks on recent public key encryption with keyword search schemes," in ATC, 2008, pp. 100–105.
[8] J. Baek, R. Safavi-Naini, and W. Susilo, "On the integration of public key data encryption and public key encryption with keyword search," in Information Security ISC, 2006, pp. 217–232.
[9] H. S. Rhee, W. Susilo, and H. Kim, "Secure searchable publickey encryption scheme against keyword guessing attacks," IEICE Electronic Express, vol. 6, no. 5, pp. 237–243, 2009.
[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," IEEE Transactions on Information Forensics and Security, vol. 10, no. 9, pp. 1993–2006, 2015.