ISSN: 0975-8585

Research Journal of Pharmaceutical, Biological and Chemical Sciences

Key-Summative Searchable Encryption (KSSE) for information distribution via Cloud Storage.

Sriman Narayana*, Thiruneelan, and Sathiyavathi.

Faculty of Computing Sathyabama University, Chennai, Tamil Nadu, India.

ABSTRACT

This paper describe a new public-key cryptosystems which produce constant-size cipher texts. Encryption keys also come with two symmetric Key or asymmetric (public) key. Using symmetric Encryption, when Alice wants the data to be begin from a third party, she has to give the encryption her secret key; obviously, this is not always desirable This compact aggregate key can be easily sent to others and stored in a smart card with very limited secure storage. We provide requirement security analysis of our design in the standard model. We also describe other application of our designs. In particular, our System give the first public-key encryption for flexible hierarchy, which was yet to be known. We solve this problem by producing a special type of public-key encryption which we call key-aggregate cryptosystem. In this users encrypt a message not only with a public key but also with an identifier of cipher text called class. That says the cipher texts are further classified into different classes. key owner holds a unique secret called unique secret key, which can be used to extract secret keys from different classes.

Keywords:

SSE – Searchable Symmetric Expression
KAC – Key Aggregate Cryptosystem
SQL – Structured Query Language
SE – Symmetric Encryption
PKES – Public Key Encryption Scheme

Corresponding author

ISSN: 0975-8585

INTRODUCTION

Our aim is to check when the sensitive data of distributor has been leaked by agents, and if possible to identify the user that leaked the data. consider where the original sensitive data cannot be commuted. commotion is very useful to change the data and make less secure before being handed to users [1]. Traditionally, leakage detection is handled by watermarking [6,7], e.g., a unique code is present in each sent copy. If that copy is later discovered by unauthorized user, the leaker can be identified. In this paper we study unobtrusive techniques for finding leakage in set of objects or records. After giving a set of objects to users, the distributor finds some of those same objects in an unauthorized place. At this point the distributor can found that the leaked data came from one or more agents. We also present algorithms for distributing objects to users, in a way that improves our chances of identifying a fault. Finally, we also consider the option of adding unauthorized objects to the distributed set. Such objects do not point to real entities but appear realistic to the users. In a sense, the false objects acts as watermark for the entire set, without changing any individual members. If it turns out a user was given one or more fake.

LITERATURE SURVEY

Data sharing is an important functionality. In this it is shown how to securely, efficiently, and freely share data with others in cloud storage [1]. The change is that one can aggregate any set of unique keys and compress them as a single key. In other words, the unique key holder can release a constant-size aggregate key for usual choices of cipher text in cloud storage, but the other encrypted files outside the set stays secure. This compact aggregate key can easily be sent to others and stored in a smart card with very limited security for storage.

It construct a searchable encryption scheme that enables keyword search with different keys [3,4]. The scheme is simple and was designed to be included in a new system for protecting data security in client server applications against attacks on the server.

Cloud computing provides an practical and efficient solution for sharing resource within cloud users [2]. Unfortunately, securing data and identity from an unsecure cloud is still a challenging problem. Meanwhile, the storage and encryption cost of our method are independent with the number of revoked users. In addition, we analyze the safety of our method with many proofs and shows the efficiency of our scheme in experiments.

Searchable symmetric encryption allows client to encrypt data in such a way that this data can be searched. The immediate application of SSE enables a client to safely outsource its data to an unsafe cloud provider without sacrificing the capacity to search. SSE has been the focus of active research and many schemes that achieve various stages of security and efficiency has been proposed. Unfortunately, none of the previously-known SSE methods achieve all these properties at the same time. This limits the value of SSE and decreases its chance in real-time cloud storage systems.

Symmetric searchable encryption the storing and the retrieving are done by the same user. Most conventional searchable encryption methods have two disadvantages. First searching the stored documents takes linear time and the size of the database and uses heavy arithmetic operations. Secondly the existing schemes do not consider flexible attackers, a search-query will give us information about documents stored in the future. If they consider this it is at a particular cost to the performance of updates [10]. In the proposed approach a novel symmetric searchable encryption method that offers searching at constant time with the number of unique keywords stored in the server. It presents two variants of the basic method which differ in the efficiency of search and storage. Proposed system shows how each scheme could be used in a personal health record system.

ARCHICECTURE MODEL

Fig 1 depicts the system Architecture and flow of user and server and interconnection of the user and server Data sharing is important in cloud storage. In this article, we show how to securely, efficiently, and usually share data with others in cloud storage. We provide new public-key cryptosystems which produce constant size cipher texts such that decryption rights are efficiently delegated for any set of cipher texts [5].

ISSN: 0975-8585

Encryption keys also come with two types symmetric Key or asymmetric key. Using symmetric Encryption [4], when user wants the data to be begins from a third party, she has to give the encryption her secret key; obviously, this is not always good This compressed aggregate key is sent to others or will be stored in a smart card with very limited secure storage. We provide custom security analysis of our designs in the present model. We also describe other application of our designs. In particular, our designs give the first public-key patient-controlled encryption, which was yet to be known. This problem can be solved with a special type of public key encryption called key aggregate cryptosystem (KAC) [2]. In KAC, encryption of a message is done by the users with a public key and also under cipher text called class. That means the cipher texts are addition categorized into different classes. The key owner holds a unique secret called unique secret key, which can be used to retrieve secret keys for different classes.

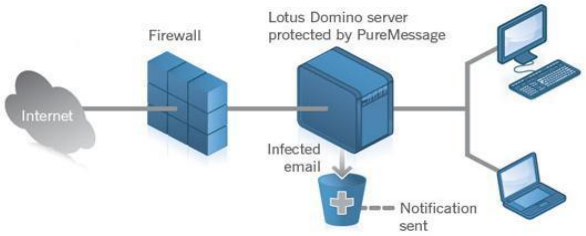


Figure 1 : System Architecture

PROPOSED SYSTEM

In proposed method we generate a tree hierarchy of symmetric keys by using block-cipher. In the proposed approach the delegation of decryption can be easily implemented with the aggregate key, which is only of fixed size. KEY-AGGREGATE ENCRYPTION gives the framework and definition for key aggregate encryption. Proposed data allocation action (across the agents) helps to improve the probability of identifying leakages.

Advantage of proposed system:

- In existing system we didn't implement the secret key generation
- The encryption needs to get the corresponding secret keys to encrypt data, which is not applicable for many applications.
- Since their method is used to produce a unique value rather than a Pair of secret keys, it is unsure how to apply this idea for public-key encryption scheme
- A limitation in our work is the predefined constrained of the number of maximum cipher text classes.
- In cloud storage the number of cipher texts usually increases rapidly, so we have to fund enough cipher text classes for the future Extension.

ORGANIZATION OF WORK

MODULE DESCRIPTION

There are 4 modules in this application listed as below.

- Data Allocation
- Fake Object
- Optimization
- Data Distributor

DATA ALLOCATION MODULE

The aim of our project is the data allocation problem that how can the distributor "intelligently" give data to a user in order to improve the chances of detecting a guilty user.

ISSN: 0975-8585

FAKE OBJECT MODULE

Fake objects are those created by the distributor in order to increase the chances of finding users that leak data. The distributor may add fake objects to the distributed data in order to increase his effectiveness in detecting guilty users. Our use of fake objects is activated by the use of trace records in mailing lists.

OPTIMIZATION MODULE

The Optimization Module is to allocate distributor data to users who has one constraint and objective. The distributor's work is to satisfy user requests by giving the number of objects they request. His objective is to be able to detect an user who leaks any portion of his data.

DATA DISTRIBUTOR MODULE

This module is designed to transfer data from distributor to users. The same module can be used for unauthorized data transfer from authorized to agents to other agents.A data distributor has given secure data to a set of trusted users. Some of the data is leaked and found in an unsafe place . The distributor must assess the leaked data came from one or more users, as opposed to gathered by other means. A data distributor needs to save the sensitive data at the time of distributing to the different users [10]. The sensitive data should be transferred in a safe way and it should not be leaked .

IMPLEMENTATION

HARDWARE REQUIREMENTS

The Processor used in hardware requirement is Intel i3, the RAM used as 2 GB, the Hard Disk used as 300 GB, the

SOFTWARE REQUIREMENTS

The Front End is used as Web Application, the Back End is used as SQL**,** the Operating system is used as Windows,

SYSTEM DEVELOPMENT

The coding is use as C# DOTNET

INPUT DESIGN

The input model is the link between the system and the user. It has the developing details and procedures for data establishing and those steps are needs to put transaction data in to a usable form by investigate the computer to read the data from a written document or it can occur by having user processing the data straightly into the system. The design of input aims on controlling the amount of input required, maintaining the errors, avoiding lag, avoiding extra steps and maintaining the process simple. The input is designed in such a way to provides security and ease use of with engaging the privacy.

OUTPUT DESIGN

A good output is one, which meets the requirements of the end user and provide the data clearly. In any system the results of the data are sent to the users and to other system by outputs. In output design it is described how the information is sent for immediate need. It is the most important source of information to the user. good and better output design improves the system to help user decision-making.

1. Designing computer output should done in an organized, well thought out manner; the right output must be developed while making sure that each output is made so that users will find the system can use easily and effectively.
2. Selected methods for providing presentation

ISSN: 0975-8585

3. Document, report or other formats that contain information produced by the system.

RESULTS AND DISCUSSION

The final outcome page consist of the followings such as application start up page which provide the user ID and the password. Once when the user login the application, few features are available such us application, file history, admin login, file action, history page.

Figure 2: This page provide user name and password this is the start up page for this application. Login page provide the personal information about the user such as name, user id, password etc.

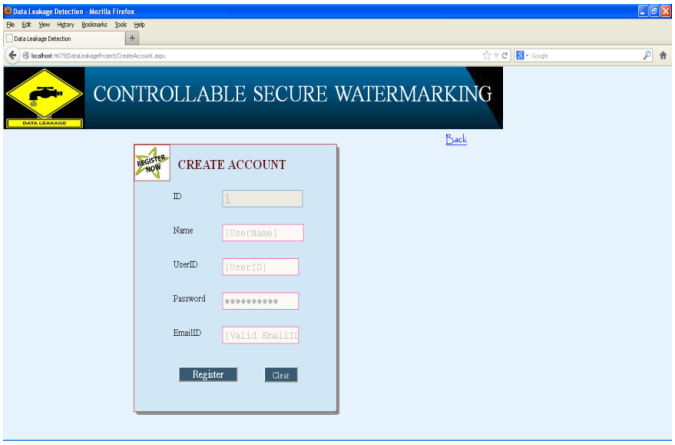


Figure 3: This page providesuserid and password you created . The id for every creation is unique.

Figure 4: This page gives us a error message if login details are entered incorrect

ISSN: 0975-8585

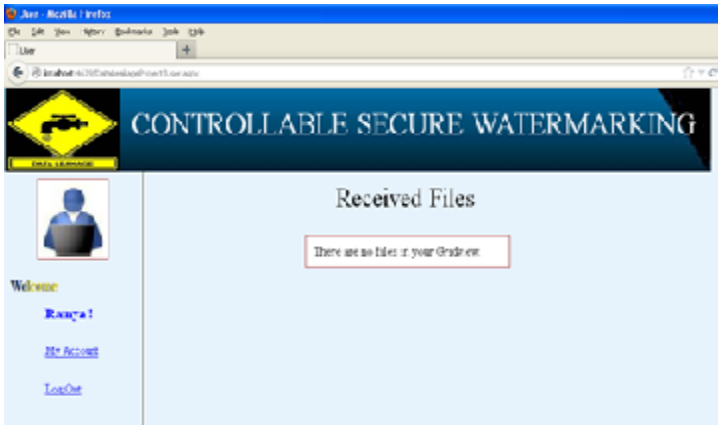
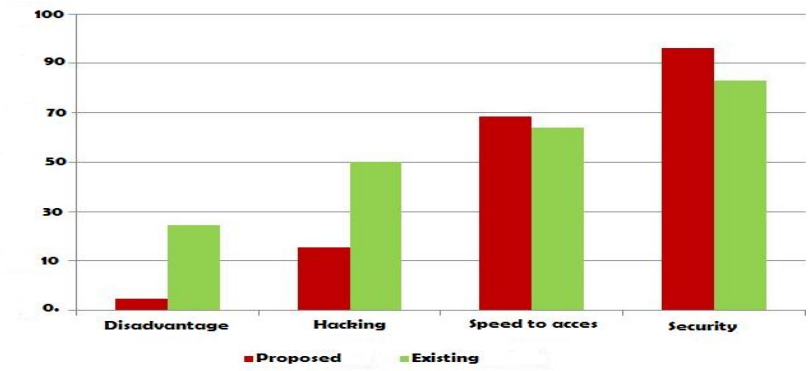


Figure 5: This is the page where the history of the received and sent can be seen by the user

Figure 6: This is the main page where the file that needed to send is Uploaded and sent to the specific user mentioned with a subject name

Figure 7: This is the page where the details of the document sent is present So that it can be used to download the file

Comparing Proposed and Existing



ISSN: 0975-8585

CONCLUSION

There is no need to provide sensitive data to users that may maliciously leak it. we should create a watermark for each object so that we could find its original place with certainty. However, in many cases we must indeed work with users that may not be 100% true, and we may not be sure if a leaked object came from an user or other source, since that data cannot admit watermarks.

We have shown it is possible to assess the user responsible for a leak, based on the combination of his data with the leaked data and the data of other users, and based on the probability that objects can be known by other means.

Our model is normally simple, but we believe it takes the essential trade-offs. The algorithms we have provided implement a variety of data distribution theories that can increase the chances of identifying a leaker.

We have shown that distributing objects can make a big difference in identifying unauthorized user, especially in cases where there is large crossover in the data that user must receive.

FURURE ENCHANCEMENT

The future work for this application would make it more adaptable and more responsive. For any application to survive in the market, it need to be updated regularly or enhancements must be made to a novel scheme for separate reversible data hiding in encrypted images. content owner encrypts the original image using an encryption key. Then a data may be compressed to least significant bits of the encrypted image using a key to create a space for some additional data. if a receiver has the data key he can get the additional data though he does not know the image content.

REFERENCES

[1] S Yu, C Wang, K Ren, and W Lou. Proc IEEE Transactions on 2014; 25 (1) : 222-233..
[2] C Chu, S Chow, W Tzeng, et al. IEEE Transactions 2014; 25(2): 468-477.
[3] R Curtmola, J Garay, S Kamara, R Ostrovsky. Journal of the ACM (JACM) 1984; 31(3) : 538-544.
[4] S Kamara, C Papamanthou, T Roeder. NY,USA: ACM , 2012; pp. 965-976.
[5] D Boneh, CG, R Ostrovsky, G Persiano. EUROCRYPT 2004; pp. 506C522.
[6] R Agrawal and J Kiernan. In Proceedings of the 28th international conference 2002; VLDB 02 : 155–166.
[7] L Mary Gladence, T Ravi. Res J Pham Biol Chem Sci 2015; 7(2) : 1274-1279
[8] P Bonatti, SDC di Vimercati, and P Samarati. ACM Trans Inf Syst Security 2002; 5(1) : 1–35.
[9] F Hartung and B Girod. 1998; 66(3) : 283–301.
[10] B Mungamuru and H Garcia-Molina. IEEE Transactions on 2011; 23(1) : 51-63