PRIVACY PRESERVING DATA ACCESS CONTROL USING PUBLIC KEY ENCRYPTION IN CLOUD

S.I Shaik Hussain#, V.Yuvaraj*

Dept. of Computer Science and Engineering, Anna University Regional Centre, Coimbatore India.
[1]shaikhussain2207@gmail.com# [2]yuviinaucbe@gmail.com*

Abstract *—*

Cloud security is one of the most important ongoing research, technology of the day to day life where the information stored in the cloud must be restricted to the unauthorized users and also it can be protected on the cloud server. In the existing Certificate less Public Key Encryption, it uses expensive pairing operations during each and every time, when a user wants to share the data in the public cloud. It is inefficient because it takes longer time to store and retrieve data in the public cloud. Hence, in this work it uses pairing free operations. The centralized public cloud used as both secure storage and also key generation centre. By considering the users public keys, the data administrator upload the sensitive information to the cloud. Finally authorized users decrypt the same encrypted information from the public cloud. This technique has greater performance and security than the earlier one.

Keywords— Certificateless encryption, Public key cryptography, Access Control

.

I. INTRODUCTION

Cloud Computing is one of the dreamt technology of utility computing. Multiple users can able to store their data remotely in the cloud. Hence, they can achieve on-demand quality services and applications from a shared pool of accountable resources. Outsourcing important information to the Storage Service Providers (SSP) will provide variable savings to the file owners in terms of both cost and security. Some of the important storage services provided by SSP are Google drive, Dropbox and iCloud to the users [4]. Naturally high sensitive data are stored in the cloud . For example, medical database and genome datasets are kept safer than the ordinary one.

Generally sharing the data in the cloud server storage is one of the most important significant functions, but usually it has many risks during the data manipulation. Because the data to be processed generally resides outside of the data admin. Even though the storage is a secured one, there may be a chance to file disclosure such that the cloud owner protect their files to a high degree of confidentiality. For encryption and decryption generally cryptography is used. There are two broad categories of cryptographic techniques such as conventional technique and public key cryptography[3]. Cheng-Kang Chu et al refers that the public key (asymmetric) encryption technique gives more confidence and flexible than conventional technique (symmetric encryption) [5][1]. In public key encryption technique, generally public key is used for encryption and private key is used for decryption.

Usually many fraudulent service providers who are already linked to the cloud storage, may look at the files which are stored in the cloud server. But it is not possible to retrieve the information, since the files are usually encrypted.. Even these files cannot be retrieved by the third party also, because of encryption. Tracking the information in the medical database and networks are usually very critical, but it needs encryption if they need to reside in the cloud.

Some of the new techniques offer information to identify the fraudulent users, retrieves the files in the cloud. Even though it provides information, it cannot clearly identify that this is a third party user. Hence, to achieve this, a specialized cryptographic technique is needed to protect the files. The asymmetric public key encryption is shown in figure 1.

In this technique, generally with the help of the cloud storage user's public key information along with their Access Control policies (ACP's), the data admin encrypts the sensitive data to the public cloud storage [7]. Once the data is uploaded and authentication takes place, a

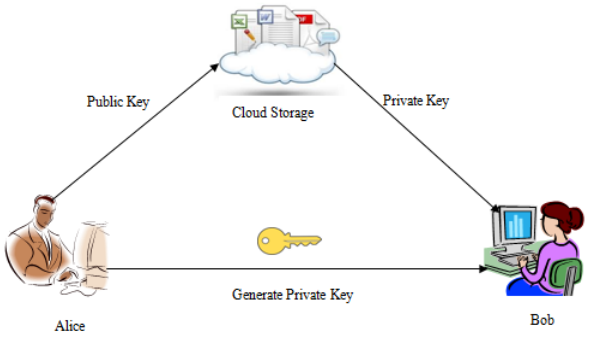

Fig 1. Asymmetric Public Key Encryption

part of the information is partially decrypted by the cloud itself. Finally, when the cloud users are in the need of data which are available in the cloud, can decrypt the data totally.

II. Related Works

The research work related to these topics is yet to be identified in the reference. Already some of the work carried forward to check whether it achieves greater performance in terms of both cost and time. We will see some of the topics related to this public key encryption.

A. Computational Diffie-Hellman Problem (CDH)

It is defined as per the following. Consider two prime numbers p and q such that q(p-1). Consider a new variable g be the greater of Z_p^*. Introduce 'A' which acts as a adversary. When 'A' comes into existence, it will solve the particular problem such that (g, g^a ,g^b) is uniformly chosen such that a,b,c $\in Z_q^*$. Let us compute k= g^{ab} . Here it defines that, A's advantage in solving the CDH problem by,

$$Adv\ (A) = Pr[A\ (g, g^a, g^b) = g^{ab}]$$

The above technique provided by Diffie Hellman to solve the particular problem of dependency between the system, when searching the data.

B. Approaches towards Encrypted Information

Generally encrypted information files are identified by some protocols, which are available in the market. For example, the particular searchable encryption technique usually kept a decryption key for the corresponding searching content available in the cloud. Curt Mola et al [11] introduce a new technique called private key storage, which provides a very limited storage capacity in the cloud. Users have greater access to large amounts of symmetrical encryption information in cheap cost [8]. The searchable content is usually called a searchable encryption. These methods address more complex queries such as conjunction, disjunction and range partition etc. Searchable symmetric encryption (SSE) allows third party to outsource the storage to another party in some other manner, where it maintains to search drastically in it. This is the most important innovative searching technology yet to be identified for the research purpose.

C. Generating Certificates along with digital signature

Boneh et al [12] produces a new technique called digital signatures, which are needed to manually generate their own key along with the signature. Specifically Low-bandwidth communication environment needs short signature[2]. Generally the short signatures are used in many different environments such as communication and postal etc. The most important widely used signatures are DSA and RSA where usually they provide long signature compared to other security methods. Here short signature scheme is used that is mainly depend on computational Diffie-Hellman, where it is already discussed in section A.

D. Effective Computational Scheme

The primary focus on the day to day life problem is security. The most important technique is two party secure computers[13][14]. It is generally used a low efficient protocol than the other computational technique. In earlier days homomorphic encryption technique is used for computation[15]. The technique is very costlier that the other one. While talking about public key encryption, numerous security threats have to be faced. This technique is IND-CCA secure which is described in the CDH problem already. Thus the IND-CCA secure provides probabilistic polynomial time to solve the particular problem in the certificateless public key encryption scheme [9]. However [16] numerous techniques show the studies on the intersection of two selected sets, which must be testable before it was selected for encryption. Usually the data is available in the ciphertext format, such that it is hidden to remain others [6]. However, we provide more secure computation using the certificateless encryption.

III. CONTRIBUTION

The public key encryption scheme consists of 7 tuples namely (Setup, Encrypt, SetPublicKey, SEM-decrypt, SetPrivateKey , SEM-KeyExtract, USER-Decrypt). The Setup function generally depicts the basic arrangements needed to do this encryption technique. SetPrivateKey is mainly used to generate the private key pair for the cloud registered users. With the help of this function the private key is generated for each and every registered user in the cloud. SetPublicKey is similar to that of SetPrivateKey , where it is used to generate the public key pair for the users available in the cloud. SEM-KeyExtract is a security mediator where the main function is to generate the SEM keys for the users and thereafter the same key is stored in the cloud storage for future decryption. Encrypt is the major function involved in this scheme, where the content of the data admin is encrypted with the help of cloud user's public key and the KGC keys of users. The encrypted data is uploaded in the centralized cloud storage along with the access control policies of the users. Now the encrypted data is available in the cloud storage.

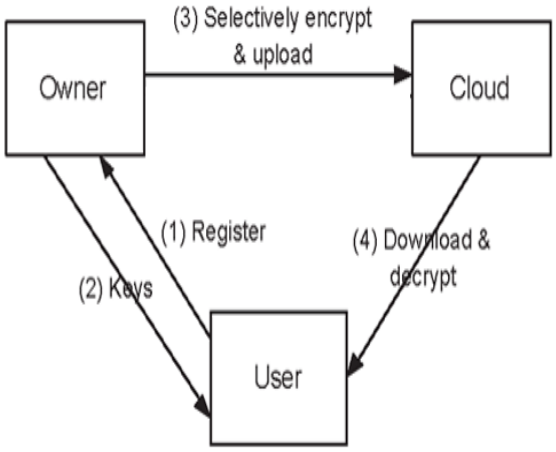

Fig 2. Overview of Certificateless Encryption

SEM-decrypt is the other function which is used to partially decrypt the encrypted data in the cloud itself. With the help of the SEM keys and the public keys of the cloud registered users, partial decryption takes place once the access control policies of the users are verified. Finally USER-Decrypt function is used to decrypt the entire content available in the cloud. Whenever a user is in the need of data from the cloud storage, the access control policies of the users are first verified and simultaneously checks the encrypted content of the particular user resides in the cloud. Upon successful authorization, the user totally decrypts the partly encrypted data available in the cloud. The basic working principle of certificateless encryption is shown in figure 2

IV. OPERATIONAL DESCRIPTION

A. *Security Token Issuance for the Cloud Users*

A token is nothing but an authentication device which is used to provide access to operate on its own. Here it provides identity in such a way that both the participating communication device is a legally authenticated one. Generally the Identity Providers (IDP's) are third parties which issues security tokens or software tokens to the cloud users based on the access control policies and their identity attributes [10][17s]. Once the IDP's have successfully issued tokens, their job is completed and henceforth there is no need for the future process.

B. *Token Registration by the Cloud Users to the Cloud Storage*

First, each and every user needs to generate their own key pair called private and public keys PK and SK, with the help of the SetPrivateKey and SetPublicKey operations respectively. Once successfully key pair has been generated the users register their public key along with the ID with the Key Generation Centre (KGC) which is available in the public cloud. For the users request, the KGC generates a public key which is unique for the users. Also, it generates two partial keys. In this one partial key is named as SEM key which resides in the cloud itself. The other partial key U-key is meant for users. The public key is nothing but the KGC, which is a combination of both user and KGC generated public keys. The KGC plays an important role, such that it encrypts the user's data. All other remaining keys are used to decrypt the data.

C. *Encryption and Uploading Information to the Cloud*

If the data owner wants to upload the encrypted content means first, data owner gets KGC keys of cloud users from the cloud storage. With the help of the KGC keys of the users and the public keys, the data owner uploads the encrypted content to the cloud. This is clearly shown by the following algorithm.

Let us consider a plaintext for encryption where M ∈ $(0,1)^n$ for the entity A with identity ID_A and public keys ($u_A, w_0, w_1 d_1$), the following action will be performed.

1) Check whether $g^{d1} = w_1 . y^{H2(ID_A , w_0, w_1)}$.

If the produced result is invalid, we cannot use encryption algorithm.

2) Choose σ € $(0,1)^{k0}$ and

Compute r = H_3(M, σ, ID_A , u_A).

3) Compute $C_1 = g^{r.}$

4) Compute $C_2 = (M||\sigma) \oplus H_4 (u_A{}^r) \oplus H_5(w_0{}^r . y^{H1(ID_A, w_0)} .r)$.

5) Compute C_3

$C_3 = H_6 (u_A, (M||\sigma) \oplus H_4 (u_A{}^r), C_1, C_2)$

Output Ciphertext C= (C_1, C_2, C_3).

In Step 1, if someone wants to encrypt a message, it can be verified by checking the validity of the user's public key. From Step 2 to Step 5 the encryption process takes place.

D. Decryption and Retrieval from the Cloud

If a user wants to retrieve the data from the cloud, it first sends a request to the SEM, which is available in the cloud to get the partially decrypted data. The SEM process two step verification, such that users ACP and the KGC- key encryption. Once successfully verified the SEM partially decrypt the encrypted content of the user. Finally, with the help of private and user key SK, U-key respectively, the user decrypts the entire partial decrypted data from the cloud. This can shown in the following algorithm.

Consider C_1 *and* C'_2 from the SEM, where A performs the following action using the private key z_A

1) Compute $C_1{}^{ZA}$

$C_1{}^{ZA} = g^{r. z_A} = g^{ZA.r} = u_A{}^r$

2) Parse M' and σ' from M'||σ' = $H_4 (C_1{}^{ZA}) \oplus C'_2$

3) Compute r' = H_3(M', σ', ID_A , u_A) and $g^{r'}$

4) Check whether $g^{r'} = C_1$

If the verification successfully happens, it returns the fully decrypted message M'=M. If the verification has not successfully happened USER-Decrypt cannot process. In Step 1 and Step 2 is used for user decryption using its own private key. Step 3 is used for validity checking of A. Finally Step 4 is used to check whether the decryption is successfully happened or not.

V. Performance Evaluation

A. Data Integrity

Generally storage server ensures the availability of data at each and every time during the retrieval by the users. It ensures high authentication for the encrypted content from the unwanted users in the cloud. By this technique we can show that the confidentiality of the data is protected within the cloud storage as well as the data is available for the trusted users in the cloud. The performance comparison of RSA with other cryptographic algorithms is shown in figure 3.

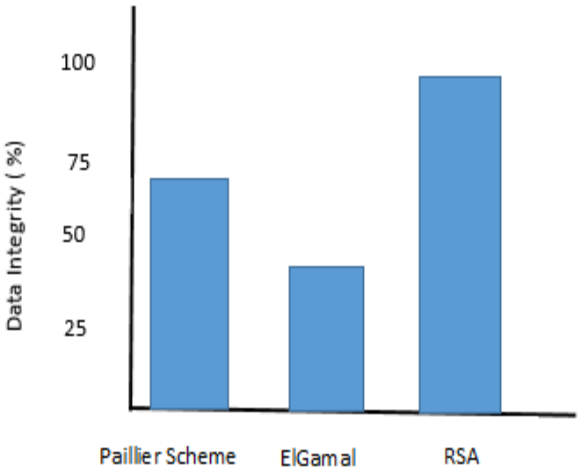

Fig 3. Data Integrity

B. Average Waiting Time

A user access the file stored in the cloud server at a particular amount of time, at the same time another user access the same file. He should wait for a time until the current user fetch and modify the file. We compare the four users to fetch and modify the same file at different time scale. The following diagram clearly depicts the comparison of four users to fetch and modify the content in the same public cloud.

Also from the diagram, we can note that the average waiting required to fetch and modify the encrypted content varies for each and every user. This is because of each and every user is in the need of their own

encrypted content which is needed to be decrypt here. The performance comparison of Average waiting time shown in fig.4

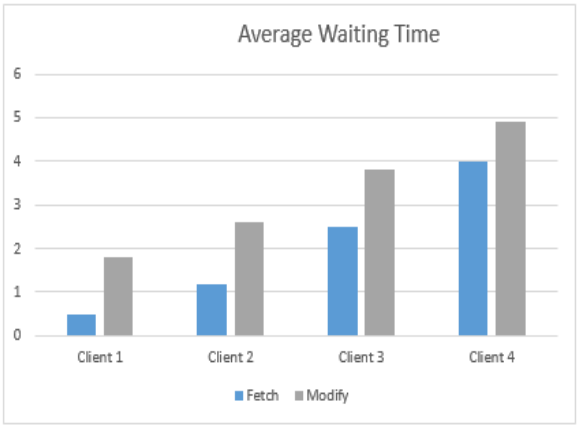

Fig.4. Performance comparison of Average waiting time.

VI. Conclusion And Future Work

In this paper, we discuss that the certificateless encryption is used to ensure the authentication for storing files. It guarantees high integrity and more security on the cloud users. This technique provides greater security by providing double encryption. Since using double encryption, the time required to encrypt the confidential data is much longer. Hence, in future we concentrate to reduce the total time required to encrypt the confidential data in the cloud storage. Also planned to test this technique using some other related algorithm for performance evaluation.

References

[1] A.Sahaiand, B.Waters, "Fuzzy identity-based encryption," LNCS 3494 in Proc.EUROCRYPT, Aarhus, Denmark,2005,pp.457–473.

[2] D.Pointcheval and J.Stern, "Security arguments for digital signatures and blind signatures," J. Cryptology, vol.13, no.3, pp. 361–396, 2000.

[3] G.Miklau and D.Suciu, "Controlling access to published data using cryptography,"inProc.29thInt.Conf.VLDB,Berlin,Germany,2003, pp.898–909.

.

[4] Cheng-Kang Chu, Sherman S.M. Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H. Deng, "Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage" IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 2, FEBRUARY 2014.

[5] J. Zhuge, T. Holz, C. Song, J. Guo, X. Han, and W. Zou. Studying malicious websites and the underground economy on the Chinese web. In Workshop on the Economics of Information Security, June 2008.

[6] J.Bethencourt, A.Sahai, and B.Waters, "Ciphertext-policy attribute-based encryption,"inProc.2007 IEEESymp.SP, Taormina,Italy,pp.321–334.

[7] J.Camenisch, M.Dubovitskaya, andG.Neven, "Oblivious transfer with access control," in Proc.16th ACM Conf.CCS, NewYork, NY, USA, 2009, pp.131–140.

[8] M. Kavitha Margret ,"International journal of advanced research in computer engineering and technology", vol.2 ,2013.

[9] S.Al-Riyami and K.Paterson, "Certificate less public key cryptography, "in Proc.ASIACRYPT2003,C.-S.Laih,Ed.Berlin,Germany: Springer,LNCS2894,pp.452–473.

[10] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation,"inProc.5thASIACCS, NewYork, NY, USA,2010,pp.261–270.

[11] R. Gennaro, C. Hazay, and J. S. Sorensen. Text search protocols with simulation based security in 13th International Conference on Practice and Theory in Public Key Cryptography, pages 332–350, 2010.

[12] Curtmola, R., Garay, J., Kamara, S., Ostrovsky, R.: Enhancing Definition and Efficient Constructions using Searchable Symmetric Encryption In: 13th ACM Conference on Computer and Communications Security, pp. 79–88 (2006)

[13] Boneh, D., Lynn, B., Shacham, H.: Generate Short signatures Using weil pairing. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 514–532. Springer, Heidelberg (2001)

[14] Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game. In: 19th ACM Symposium on Theory of Computing, pp. 218–229 (1987)

[15] Yao, A.C.: Protocols for secure computations. In: 23rd IEEE Symposium on Foundations of Computer Science, pp. 160–164 (1982)

[16] C. Gentry. Fully homomorphic encryption using ideal lattices. In 41st ACM Symposium on Theory of Computing, pages 169–178, 2009.

[17] V.Goyal, O.Pandey, A.Sahai, and B.Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc.13th ACMConf.CCS, NewYork, NY, USA, 2006, pp.89–98