International Journal of Pure and Applied Mathematics
Volume 117 No. 15 2017, 1143-1149
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

Fully Homomorphic Public Key Encryption Based on Arbitrary Key Aggregation Cryptosystem (AKAC) for Secured Data Communication in Cloud Infrastructure

[1]Dr.M.Sayee Kumar, [2]Dr.G.M.Karthik and [3]A.Aruna

[1,2]Assistant Professor, School of Computing, SRM University, Kattankulathur, Kanchipuram Dt.
[3]Assistant Professor, SRM University, Ramapuram, Chennai.

E-mail:[1]sayeekumar.m@ktr.srmuniv.ac.in, [2]karthik.gm@ktr.srmuniv.ac.in and
[3]mailtoaru25@gmail.com

ABSTRACT- Cloud computing is an emerging technology where it provides multiple services like platform, infrastructure, security, and storage. To be precise Cloud storage can provide the prosperity of greater accessibility and reliability; rapid deployment; strong protection for data. The data encryption is done to ensure the security. Data sharing is the practice of making data used for scholarly research available to other investigators. In this proposed protocol the method of sharing encrypted data securely, efficiently with others in cloud storage are taken as the major aspect. A blueprint of an Arbitrary Key aggregated cryptosystem (AKAC) based on fully homomorphic encryption is proposed in which relatively small key is computed and ciphertext size is notably small. The public and private keys consist of two large integers are used in the construction of fully homomorphic scheme and similarly, the cipher text consists of one large integer are evaluated. As such, this scheme has smaller message expansion and key size which is secure under key dependent encryptions.

Keywords: Fully homomorphic encryption, Cloud storage, Data privacy, Arbitrary Key Aggregation Cryptosystem (AKAC), Aggregation key.

1. INTRODUCTION

Cloud computing is the process of delivering computing services over the internet. Cloud services provide the user to use both software and hardware which is managed by the remote server. Examples of cloud services which include online file storage, social networking, and online business applications. In general, the Cloud computing is for anything that involves remit hosted services over the Internet. These administrations are separated into three sorts: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). The name cloud computing was motivated by the cloud symbol that is regularly used to represent the Internet in flowcharts and diagrams.

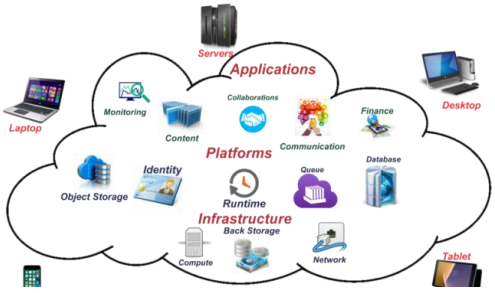

Fig.1.1 Architecture of Generic Cloud Computing.

Data sharing is a crucial functionality. For example, Bob's store data in an encrypted format and he wants to share those data to Alice. If he extracts those encrypted data and decrypts it and sends those original data to Alice, there is no use of cloud storage. Otherwise, he needs to send the decryption key to Alice, which loses the data owner's privacy.

Therefore, the best solution for this is that Bob encrypts all data with different keys and send the single aggregated key to Alice, which allows Alice to process on that data without retrieving original data and decryption key. It is designed by the scheme called Fully homomorphic encryption.

2. RELATED WORK

In cryptography, encryption is the process of encoding messages or information in such a way that only authorized parties can read it. Encryption may be symmetric or asymmetric encryption. Symmetric key will make use of the same key for both encryption and decryption. Where asymmetric encryption will use different keys for encryption and decryption. Our approach is purely based on asymmetric encryption.

J. Benaloh *et al* have proposed an electronic health record system in which all the health record has been stored in encrypted form. This system allows each patient to generate their own decryption key. this refers to Patient Controlled Encryption (PCE), Where this patient to share their record with doctors and health care providers. The system design is based on a hierarchical encryption system. Each record is partitioned into a hierarchical structure. The patient is supposed to store a root key, from which a tree of sub keys is derived. The patient can share sub keys for decryption of different portions of the record.

2.1 Compact key in attribute-based encryption

Attribute-based encryption is the encryption which is based on fewer attributes of customer or user.(For example, User Name, Date of birth etc,) V. Goyal et al (2006) have proposed a new crypto system for sharing encrypted data by fine-grained level. This scheme is called a Key-Policy Attribute-Based Encryption (KP-ABE).In this system, each ciphertext is named with a set of attributes. It controls which ciphertext can be decrypted by a user. This scheme is based on Hierarchical Identity-Based Encryption (HIBE).But a set of attributes are not hidden under the encrypted data.

Melissa Chase *et al* have proposed Attribute-based encryption to provide privacy and security to user data. This system is based on user attributes. They have used Global Identifier (GID) and Central authority (CA) in ABE scheme.CA is to provide access to authenticated data.GID allow the user to provide their attributes.

2.2 Compact keys in Identity-based encryption

An Identity-based encryption (IBE) is discussed, Where Encryption is fully based on user identity (For example, User email ID).S.S.M. Chow et al (2012) have proposed a system to support dynamic secure cloud storage with authentication. This design is based on verifier-local revocable group signature and Identity-based broadcast encryption with fixed size ciphertext and private key. When there is dynamic user it is important to support data provenance that maintains who created, deleted, modified the stored data in the cloud. Before the user storing their encrypted data he/she has to sign those encrypted data with signing key. The ciphertext has been accepted if and only if the signature is valid.

2.3 Cryptographic key assignment for predefined hierarchy

Cryptographic key assignment (Ateniese.G, A.D et al and Akl.S.G et al) is the scheme which used for reducing the storage cost and management cost of the keys used in cryptography. For example, we can take a tree structure to represent the hierarchical key delegation. First cipher text classified into the various node. Each node has an individual decryption key. And root node contains a key for decrypting entire tree. So here partial access to a data can be made by providing key according to the specified node.

Table 1: Comparisons between Our AKAC scheme and other Schemes

Various Scheme	Decryption Key size	Ciphertext size	Encryption Type
Symmetric-Key Encryption with Compact Key	Constant	Constant	Symmetric key
IBE with compact key	Constant	Non-constant	Asymmetric key
Attribute-based encryption	Non-constant	constant	Asymmetric key
KAC	Constant	constant	Asymmetric key
AKAC	Non-constant	Non-constant	Asymmetric key

3. EXISTING METHODOLOGY:

In previous work, Key Aggregated cryptosystem (KAC) has been used to aggregate multiple keys to making as a single key. It consists of five polynomial time algorithm. Setup (1,n): the First sender has to mention the number of ciphertexts initially by using *setup(1,n)*.Where n is the number of ciphertexts to send. It consists of following five polynomial algorithms.

- *Setup*: Initial algorithm which is executed by the data holder to setup an account on an untrusted server
- *KeyGen*: This algorithm is executed by the data owner to arbitrarily produce a public/master-secret key pair
- *Encrypt*: It is executed by a person who wants to encrypt data
- *Extract*: It is executed by the data owner for delegate the decrypting power for a definite set of cipher text
- *Decrypt*: executed by a person who received an aggregate key KS which is generated by Extract

4. PROPOSED METHODOLOGY

The ultimate aim of this system is to overcome the limitation of existing work in which the number of maximum ciphertexts classes is bounded. So fully homomorphic scheme is proposed. This scheme consists of following polynomial time algorithms.

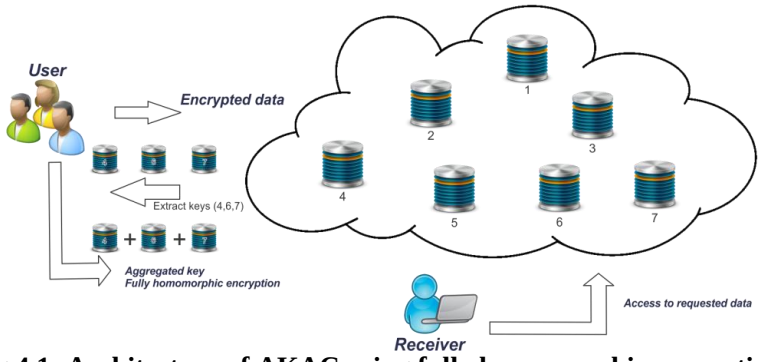

Fig.4.1: Architecture of AKAC using fully homomorphic encryption.

Table 2: Number of encryption and decryptions in various encryption schemes

Various Schemes	No. Of Encryption	No. Of .Decryption
Symmetric Key encryption	1	1
Identity based encryption	1	1
KAC	1	1
AKAC	**2.[Re-encrypt(optional)]**	1

4.1 MODEL DESCRIPTION

4.1.1Construction of AKAC

Initially, the basic network model for cloud has been created. And entities for cloud environment is created. Those different entities are User: which stores a large amount of data and process those data in the cloud. Cloud storage server: which is managed by cloud service provider (CSP), has significant storage space.Receiver: who receives an aggregate key for decryption.

4.1.2 Key Aggregation based on FHE

Fully homomorphic encryption produces relatively small key and ciphertext size. The construction follows a fully homomorphic scheme. The public and private keys consist of two large integers (one is shared by both the public and private key) and the cipher text consists of one large integer.

4.1.3 Sharing encrypted data

User or data owner will share their data in a confidential and selective way. Sharing confidential key over the internet will be risky which may be hacked by a third party or unauthorized person. But here even if the secret key has been hacked by the unauthorized person they cannot do anything with it. Because Fully homomorphic scheme maintains data integrity. Here decryption key for multiple data is shared as single aggregated key. So that key allows the receiver to process data in a confidential way.

4.1.4 Decrypt data

The single secret key has been sent to the user by which he/she can able to decrypt corresponding cipher text classes. That Decryption mechanism has been done by a receiver who receives the decryption key. Since it is Asymmetric encryption, it makes use different keys for encryption and decryption. The user can make use of decryption algorithm to decrypt cipher text classes.

4.2 ALGORITHM

Arbitrary key aggregated cryptosystem using Fully homomorphic scheme has consisted of 4 polynomial algorithm and Re-encrypt algorithm (optional).

KeyGen: Client will generate a couple of secret keys namely Public key(PK) and Secret key(SK) for encryption. It takes security parameter λ_s as an input and outputs a PK and SK, PK defines Plaintext and ciphertext spaces.

$(PK, SK) \leftarrow KeyGen_f$

Encrypt: Executed by the one who wants to encrypt the data. It takes Public Key(PK) and Plaintext P as an input Where $P \in \mu$.It gives output as ciphertext C where $C \in \Phi$.

$\Phi_i \leftarrow Encrypt_f\ (PK, \mu_i)$, Where $\mu_i = (\mu_1, \mu_2 \ldots \mu_n)$ and $\Phi_i = (\Phi_1, \Phi_2 \ldots \Phi_n)$.

Evaluate: Fully homomorphic scheme has a well-organized algorithm called Evaluate which is suitable for any public key (PK) and for ciphertexts C obtained from Encrypt. It takes input as a Public key (PK), C_i, Φ_i Which is a tuple of cipher texts. It gives output as Φ.

$\Phi \leftarrow$ Evaluate (PK, C_i, Φ_i) Where $\Phi_i = (\Phi_1, \Phi_2 \ldots \Phi_n)$.

Decrypt: It is executed by the person who receives decrypt key from the sender. Where Φ is an Evaluated Ciphertext and SK is the aggregated secret key and produces the Ciphertext μ as an output.

$\mu_i \leftarrow Decrypt_f(SK, \Phi_i)$

Re-encrypt(c, PK): At a higher level, it makes use of Re-encrypt Algorithm. It takes "dirty ciphertext" Φ as an input, removes the error produce new ciphertext $\Phi_{new.}$

$\Phi_{new} \leftarrow$ Re-encrypt (PK, Φ)

5. CONCLUSION

The fully homomorphic approach describes how to "compress" secret keys in public-key cryptosystems. This support delegation of secret keys for different cipher text classes in cloud storage using fully homomorphic public key encryption. This scheme produces relatively small message expansion and key size. There is no predefined number of maximum cipher-text classes. So the user can send many numbers of cipher text classes as the user wants to send. At a higher level, we can make use of a Re-encrypt algorithm which produces new ciphertext and dirt in older ciphertext has been removed. So this approach will provide efficient and flexible key delegation.

REFERENCE

[1] Ateniese .G, K. Fu, M. Green, and Hohenberger G, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," Proc. Network and Distributed Systems Security Symp. (NDSS).

[2] Ateniese.G, A.D. Santis, A.L. Ferrara, and B. Masucci, "Provably- Secure Time-Bound Hierarchical Key Assignment Schemes,"J. Cryptology, vol. 25, no. 2, pp. 243-270, 2012.

[3] Akl.S.G and P.D. Taylor, "Cryptographic Solution to a Problem of Access Control in a Hierarchy," ACM Trans. Computer Systems, vol. 1, no. 3, pp. 239-248, 1983

[4] Basel Alomair, Radha Poovendran," Information Theoretically Secure Encryption With Almost Free Authentication" Proc. ACM Conf. Computer and Comm. Security (CCS '06), pp. 89-88, 2008.

[5] Benaloh. J, Chase.M, Horvitz E, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records," Proc. ACM Workshop Cloud Computing Security (CCSW '09), pp. 103-114, 2009.

[6] Boneh. D, Boyen.X and Goh. E, “Hierarchical Identity Based Encryption with Constant Size Cipher text,” Proc. Ann. Int’l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT), pp. 440-456, 2005.

[7] Boyang Wang, Ming Li, Hui Li, “Storing Shared Data On The Cloud Via Security Mediator” IEEE transactions on parallel and distributed systems, vol. 15, no. 2, July 2013.

[8] 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.

Rajesh, M., and J. M. Gnanasekar. "Annoyed Realm Outlook Taxonomy Using Twin Transfer Learning." International Journal of Pure and Applied Mathematics 116 (2017): 547-558.

Rajesh, M. & Gnanasekar, J.M. Wireless Pers Commun (2017),https://doi.org/10.1007/s11277-017-4565-9

Rajesh, M., and J. M. Gnanasekar. "GCCover Heterogeneous Wireless Adhoc Networks." Journal of Chemical and Pharmaceutical Sciences (2015): 195-200.

Rajesh, M., and J. M. Gnanasekar. "CONGESTION CONTROL IN HETEROGENEOUS WANET USING FRCC." Journal of Chemical and Pharmaceutical Sciences ISSN 974: 2115.

Rajesh, M., and J. M. Gnanasekar. "GCCover Heterogeneous Wireless Ad hoc Networks." Journal of Chemical and Pharmaceutical Sciences (2015): 195-200.

Rajesh, M., and J. M. Gnanasekar. "CONGESTION CONTROL USING AODV PROTOCOL SCHEME FOR WIRELESS AD-HOC NETWORK." Advances in Computer Science and Engineering 16.1/2 (2016): 19.

RAJESH, M. "TRADITIONAL COURSES INTO ONLINE MOVING STRATEGY." The Online Journal of Distance Education and e-Learning 4.4 (2016).

Rajesh, M. "Object-Oriented Programming and Parallelism."

Rajesh, M., K. Balasubramaniaswamy, and S. Aravindh. "MEBCK from Web using NLP Techniques." Computer Engineering and Intelligent Systems 6.8: 24-26.