Authenticated Cloud Data Service Using Attribute Based Encryption

Rakesh Venkat Battu*, Mangalagowri R
Computer Science and Engineering, SRM University, Chennai
***Corresponding author: E-Mail: rakeshvenkatbattu@gmail.com**

ABSTRACT

Cloud third parties have established their existence in Cloud computing since the need to facilitate the selection of cloud services and management among users. Cloud third parties provide the brokerage schemes in the assumption that they are fully trusted and do not guarantee on the quality or correctness of the delivered services. So there are chances that the recommendation given by the brokerage can be wrong or misguiding the consumer. Hence this paper helps cloud users to securely store their data in a semi trusted cloud service provider using a secure data system and secretly share the data with selected users. This paper also works on the field of key management using ABE Encryption scheme where the data owner has the least or nothing to do with key management process. Moreover, of any user wants to access the secure file then a request is sent to the data owner. The data owner has the complete Access Control to provide authority to users. If the data owner wants to share the file then the secret key is shared with the data receiver. Now the data receiver can decrypt and download the file from the cloud server.

KEY WORDS: Cloud computing, ABE, Support Vector Machine (SVM).

1. INTRODUCTION

In cloud it's a common practice to encrypt the data before it is been stored in cloud storage. Access control is another added security upon the encrypted data to be strongly stored in the cloud platform. Attribute Based Encryption- ABE is finding its existence in cloud technology since it can deliver data privacy with one-to-many, fine grained and non-interactive access control. CP-ABE- Cipher text-policy attribute based encryption is a modified version of ABE with more flexibility in delivering security for general applications. As visualized in the below figure (Fig.1) Cloud service provider (CSP) manages all the servers and services provided to the client. The data owner can encrypt, decrypt, secure, store and share data to CSP. The files stored in cloud platform follow a hierarchical structure. A set of files will be divided into several hierarchy subgroups with different access levels. When the files in the same hierarchy group are encrypted using integrated access structure then the storage cost and time consumed for encryption is reduced. Let's consider Personal Health Record (PHR) as an example in our study. So in order to store PHR in cloud storage the PHR information M is divided into two parts namely: m1- which contains generic details like name, age, telephone number, address, etc. The other part is m2- which contains medical information of the patient like, disease, allergy, test results, medication details, operation notes, etc. Now CP-ABE is implemented on the divided information m1 and m2 using different access policies depending on the need. Cloud brokerages play the role of assisting the user to select the right cloud service. But these brokerages do not promise a trusted recommendation. In this proposed paper the data owner has the complete access control and if decided to give access to the requestor then the secret key is shared with the user. The data receiver will be able to decrypt and download the original file using the secret keys shared by the data owner. The point of concern here is that the file can be decrypted only if the secret key is same as the "matching attributes", where this key is issues by a trusted third party. There are several hierarchical CP-ABE schemes in existence. Our method of combining IBE and CP-ABE keeps the length of the secret key to minimum length with order of the attribute set. In the hierarchical ABE scheme the parent authorization domain will monitor the child authorization domain where the top most authorization domain will create the secret key for the next level domain. Hence the secret key generation is distributed along various authorization domains and hence the work load of key generation is distributed.

Cloud Computing: Cloud computing is the concept of virtualizing the physical components involved in the process of computing and storage. It is also called as Internet-based computing where resources and processors are shared to users on demand. Cloud computing offers storage solutions to enterprises and minimizes the use of physical components and management and maintenance involved. It is also said that cloud platform can offer faster applications to enterprises. The scalability feature of cloud helps in unexpected fluctuation in administration. The "pay as you go" concept of cloud computing can lead to huge pricing model if the resources are not well utilized by the administrators. With a huge choice of services like storage devices, low cost computation, high capacity networks, firewalls, software, applications, hardware virtualization, service provider and many more has led to this increase growth in cloud technology. Hence cloud technology has become the choice of many enterprises and IT companies due its ease of.

Existing System: As a demand for a brokerage agent in cloud computing, cloud brokerages are involved as an additional computational layer in assisting the user with cloud selection and service management. But the present cloud brokerage schemes are not efficient and reliable as they do not agree to deliver the correctness in the service recommended. Cloud Service Selection Verification scheme (CSSV) along with index structure (MMB cloud-tree) helps in the detection of misbehavior in the process of service selection. Simulation results show that our proposed approach proves accuracy and efficiency both theoretically and empirically.

Review of literature:

RSA (Rivest-Shamir-Adleman Algorithm): RSA- Rivest-Shamir Adleman algorithm is the most preferred one in terms of public key cryptosystem. It is the most commonly used public key scheme in the world of modern computers and technology. It is asymmetric and uses integers like 1,024 bits in size. It uses block cipher and with one round of encryption. RSA is used for both encryption and decryption. It uses two different keys for encryption and decryption and hence termed as asymmetric. This algorithm is also called as public key cryptography as one key can be shared to all and another is kept private. RSA came into existence in 1978 by Ron Rivest, Adi Shamir and Leonard Adleman. RSA gives a return product as two large prime numbers (P*Q) and (I) as the public key. The value (P*Q) is kept as the secret key. It is possible that anyone having the public key can encrypt the data but only if the prime factor of that large number can be found the user will be able to decode the message. RSA is used for digital signatures and public key encryption as well. The complexity of the algorithm depends on the complexity in factoring large integers.

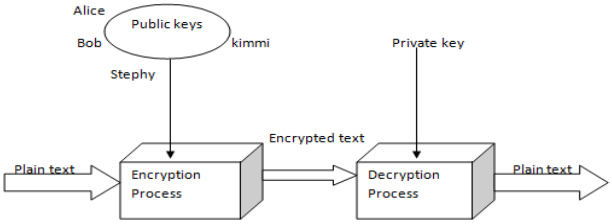

Figure.1. RSA Algorithm (Asymmetric Key Cryptography)

The below algorithm is used in RSA,

Choose p and q

Calculate n = p * q

Find φ(n) = (p - 1) * (q - 1)

Select e such that 1 < e <φ(n) and e and n are co-prime.

Compute a value for d such that (d * e) % φ (n) = 1.

Public key is (e, n)

Private Key is (d, n)

For encryption C=me (mod n) and decryption m =cd (mod n)

AES (Advanced Encryption Standard): AES is an advanced version of DES. In 1997, the National Institute of Standards and Technology (NIST) wanted to have a successor of DES. Among the several proposals, Advanced Encryption Standard was chosen and replaced DES and 3DES. AES was developed by Vincent Rijmen and Joan Daeman in 2001. AES uses symmetric block cipher structure with three block ciphers and with several rounds of encryption. AES is used by U.S government to secure classified information and is being used in the software and hardware for sensitive data encryption. AES uses AES-128, AES-192 and AES-256 block ciphers. All the block ciphers encrypt and decrypt data in blocks of 128 bits using cryptographic keys of 128 bits, 192 bits and 256 bits, respectively. There are 10 rounds of encryption for 128 bit keys, 12 rounds for 192 bit keys and 14 rounds for 256 bit keys.

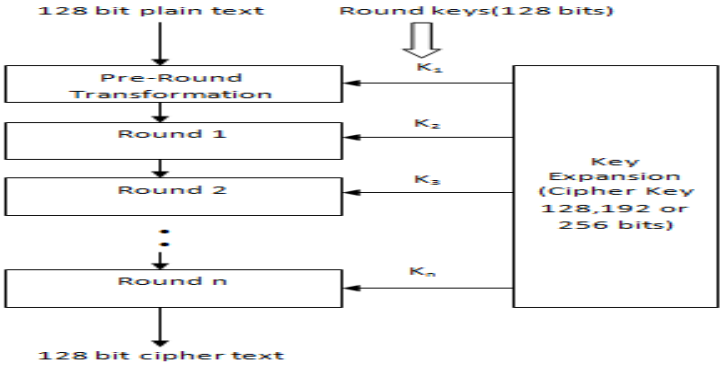

Figure.2. AES

The same process is followed in all the rounds and the encryption process can be further can be listed as four steps which includes; Substitute byte, Shift rows, Mix column and Add round key.

Substitution round: In this step the sub-bytes are substituted with byte-by-byte in the forward encryption process.

Shift Rows: The rows in the state array are shifted in this step during forward process (S-Box process).

Mix Column: The bytes in each column are mixed up during forward process.

Add Round key: The round key is added to the output received from the previous step during the forward process. This step depends on the key size.

Different round keys are used in the AES encryption process. These keys are applied on an array of data using mathematical operations. After which the data is represented in an array form called the state array.

The encryption process is detailed below:

- Round keys are derived from the cipher key.
- State array is formed using the block data or plain text.
- Round key is added to the initial state array.
- The manipulations are continued till 9th round.
- After the 10th round we derive to the final output with a cipher text.

Through the above process we get the final cipher text or encrypted text as the output.

Blowfish: Bruce Schneier introduced Blowfish algorithm in the year 1993. It's a symmetric block cipher using variable length key starting from 32 bits up to 448 bits. The block size is fixed to 64 bits. It consists of 16 rounds of Feistel cipher and uses large key dependent S-Boxes. Each S-box constitutes to 32 bits of data.

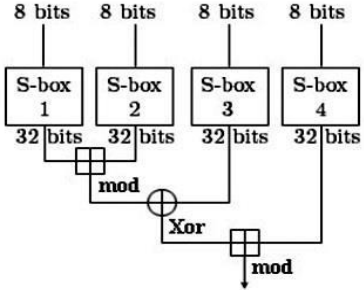

Figure.3. Blowfish

The above diagram demonstrates Blowfish's F-function. The input data of 32 bit is divided into four 8-bit quarters and is used as input to S-boxes. The outputs (Mod) modulo232 and XOR are added to derive the final 32-bit output which is the encrypted data. The same procedure in reverse order is conducted for decryption process.

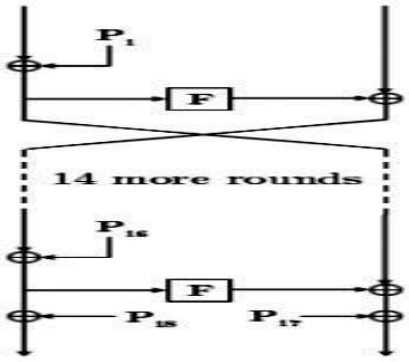

Figure.4. The Feistel structure of Blowfish

There are no attacks and loop holes found in Blowfish algorithm. This algorithm comprises of two parts namely, key expansion and data encryption. Data encryption holds 16 rounds of feistel network. Each round has key dependent permutation in P-Box and key/data dependent substitution in S-Box. Blowfish algorithm consists of S-Box and P-Box. Let's consider the P-array first, it consists of 18 keys of 32 bit and named as P1, P2, P3,...., P18. 4 S-boxes with 32 bit and 256 entries each.

S1[0], S1[20],..., S1[255]; S2[0], S2[20],..,, S2[255]; S3[0], S3[20],..., S3[255]; S4[0], S4[20],..,, S4[255].

Encryption is done using the below steps.

Divide x into two 32-bit halves:
(xL) and (xR)
For i = 1 to 16:
xL = xL XOR Pi
xR = F(xL) XOR xR
Swap xL and xR
Next i
Swap xL and xR (or Undo the last swap.)
xR = xR XOR P17
xL = xL XOR P18
Recombine xL and xR.
Divide xL into four eight-bit quarters: a, b, c, and d
F(xL) = ((S1,a + S2,b mod 232) XOR S3,c) + S4,d mod 232

Decryption is similar to encryption but with a reverse process. Blowfish has good encryption rate in software and much faster than IDEA and DES.

Twofish: Two fish also follows the feistel structure with symmetric block cipher. Bruce Schneier was the founder of this algorithm in 1998. Twofish is proved to be efficient in software's which run on smaller processors like the smart cards embedded in hardware. The implementers have the leverage to customize the encryption speed, code size and key setup time in order to optimize the performance. Twofish algorithm is un-patented, free of cost and license free. This algorithm uses key sizes of 128, 192 and 256 bits. Block size of 128 bit is used with 16 rounds of encryption.

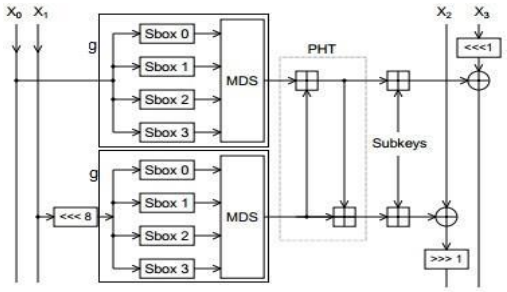

Figure.5. TwoFish

The above figure demonstrates the round function involved in Twofish algorithm. This round function happens for 16 times and derives to a final result of cipher text after the 16th round.

The process visualized in the figure is explained here.

- X0 and X1 are the inputs to the function g after the rotation by 8 bits.
- The function g is made up of 4 bytes key-dependent S-boxes followed by linear mixing step (MDS matrix).
- The results from the two g functions are combined using PHT (Pseudo-Hadamard Transform).
- After which the two keywords are added. One keyword is rotated by 1 bit and the keywords are XORed to the result on the left.
- In the next round the right and left halves are swapped.
- After the 16th round, the last swap is reversed and the four keywords are XORed with another four keywords to have the final encrypted text or cipher text.

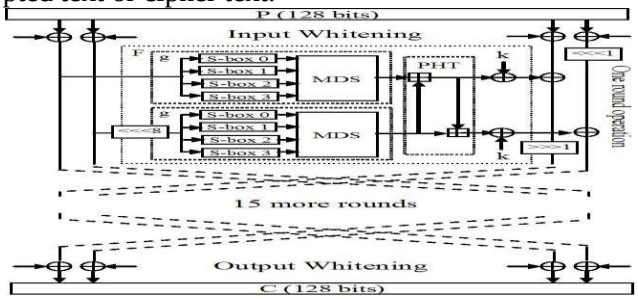

Figure.6. The Two fish Round Function Diagram

The above diagram depicts the Twofish algorithm. It contains 16 rounds of data encryption and derives a 128 bit cipher text as the final output. Twofish algorithm results in good security level but it lacks behind in encryption speed when compared with the Blowfish algorithm.

Related Work: A new method for RIBE and proposed RIBE patterns with a constant/fixed number of private key elements, we have achieved the following results.

We first devise a new method for RIBE that joins a various leveled IBE (HIBE) pattern and a public-key broadcast encryption (PKBE) pattern utilizing multi-linear maps. As opposed to the past method for RIBE, our system utilizes a PKBE pattern in bilinear maps for denial to achieve short private keys and update keys. By following our new strategy for RIBE, we propose a RIBE pattern in three-leveled multi-linear maps that consolidates the HIBE pattern of Boneh and Boyen (EUROCRYPT 2004) and the PKBE pattern of Boneh, Gentry, and Waters (CRYPTO 2005). The private key and update key of our pattern have a consistent number of group elements.

We initiate outsourcing division into IBE and propose a revocable pattern in which the revocation operations are delegated to CSP with the guide of KU-CSP, the Transactions on Computers format is full highlighted: it understands steady competence for both calculation at PKG and private key size at consumer.

User wishes not to drop a line to with PKG in key update, in another words, PKG is allowed to be disconnected after transfer the revocation list to KU secured channel or user verification is required in key-redesign amongst user and KU. We consider catching revocable IBE under a stronger opposition model. We represent an advanced development and exhibit it is secured under RDoC model, in which at least one of the KU implicit to be honest. Therefore, if a revoked user and each of the KU-CSPs intrigue, it is not able to help such user capability. At last, we offer general speculative results to reveal the competence of our expected construction.

An IBE pattern with productive revocation, whose intricacy of key updates is altogether decreased (from linear to logarithmic in the number of users), compared with the past solutions. We examined a few variations

accomplishing distinctive levels of security and also discussed about how to develop an attribute-based encryption pattern with proficient revocation. Our patterns ought to be especially helpful in the settings where a substantial number of users are involved and scalability is an issue.

The strategy has chosen cipher-text security in the random oracle model demonstrating an elliptic curve variant of the computational Difie-Hellman issue. Our system depends on the Weil pairing. We give exact definitions for secure identity-based encryption patterns and give a few applications for such systems.

The result was obtained by sharing the key generation procedure of a 2-level HIBE system from the commutative-blinding family" (started with the first scheme). As another delay, similar thoughts make it possible to construct revocable identity-based broadcast encryption patterns (utilizing the latest Boneh-Hamburg constructions).

An example in the selective-ID model. An open issue is to devise adaptive-ID secure R-IBE systems with a more tightly diminishment than what we could get. It would also be intriguing to perceive how revocation can be taken care in the context of hierarchical IBE, where every entity of the hierarchy ought to be in charge of revoking its children.

The idea of Fuzzy Identity Based Encryption, which allows for error-tolerance between the identity of a private key and public key used to encrypt a cipher-text. We portrayed two practical applications of Fuzzy-IBE of encryption utilizing biometrics and attribute-based encryption. We exhibited our construction of a Fuzzy IBE pattern that utilizations set cover as the distance metric between identities. At last, we proved our pattern under the Selective-ID model by diminishing it to an assumption that can be seen as a modified version of the Bilinear Decisional Diffie-Hellman assumption. Additionally, a Fuzzy-IBE pattern that conceals public-key that was utilized to encrypt the cipher-text is intriguing. Our pattern utilizes set-overlap as a similarity measure between identities. An open issue is to build other Fuzzy-IBE pattern that utilizes different distance metrics between identities.

2. CONCLUSION

In order to securely store and share confidential data in cloud computing we have proposed a cloud-based secure data system. Also to reduce the complexity involved in key management we have used ABE encryption scheme where the user can encrypt his/her private data. Another advantage of this proposed method is that the data owner has the complete Access Control. In this case when a user wants access to the private data then a request is sent to the data owner. The data owner will decide upon granting access and will share the secret key to the data receiver. The data receiver will use the secret key and decrypt the data and will be able to download the original data. This method is well secured and the storage providers provide secure protection to the encrypted data.

REFERENCES

Alexandra Boldyreva, Vipul Goyal, Virendra Kumar, Identity-based Encryption with Efficient Revocation, ACM, 2008, 417-426.

Amit Sahai and Brent Waters, Fuzzy Identity-Based Encryption, Cramer R (Ed.), EUROCRYPT 2005, Springer Berlin Heidelberg -LNCS, 2005, 457–473.

Benoit Libert and Damien Vergnaud, Adaptive-ID Secure Revocable Identity-Based Encryption, Spring Berlin Heidelberg-LNCS, 5473, 2009, 1-15.

Benoit Libert, Jean-Jacques Quisquater, Efficient revocation and threshold pairing based cryptosystems, ACM, 2003, 163-177.

Berlin Heidelberg –LNCS, Encryption, History-Free Update, Security Against Insiders, and Short Cipher texts, 9048, 2015, 106–123.

Dan Bonehand Matt Franklin, Identity-Based Encryption from the Weil Pairing, Springer Berlin Heidelberg-LNCS, 2139, 2001, 1-15.

Eiichiro Fujisaki, Tatsuaki Okamoto, How to Enhance the Security of Public-Key Encryption at Minimum Cost, Springer Berlin Heidelberg –LNCS, 1560, 1999, 53-68.

Jae Hong Seo and Keita Emura, Revocable Identity-Based Encryption Revisited, Security Model and Construction, Springer Berlin Heidelberg-LNCS, 7778, 2013, 216-234.

Seunghwan Park, Kwangsu Lee, and Dong Hoon Lee, New Constructions of Revocable Identity-Based Encryption from Multilinear Maps, IEEE, 2015, 1564-1577.

Tejaswi U, Vinay Kumar VPS, Identity-based Encryption with Outsourced Revocation in Cloud Computing, IEEE, 2013, 425-437.