ISSN (Online): 2319-8753
ISSN(Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization *Volume 4, Special Issue 5, April 2015*

International Conference On Emerging Trends in Engineering and Technology (ICETET'15)

On 13th & 14th March 2015

Organized by

Pandian Saraswathi Yadav Engineering College, Arasanoor, Sivagangai, Tamilnadu, India

Leakage-Resilient cryptosystems for Scalable Data Sharing in Cloud Storage

A.Monisha [1], M.Kumaresan[2]

Student, Department of Computer Science and Engineering, PGP College of Engineering and Technology, Namakkal, India[1]

Assistant Professor, Department of Computer Science and Engineering, PGP College of Engineering and Technology, Namakkal, India[2]

ABSTRACT-Data sharing is an important functionality in cloud storage. In this previous work, we show how to securely, efficiently, and flexibly share data with others in cloud storage. The existing work presents the Key-Aggregate Cryptosystem used for conveniently sent to others or be stored in a smart card with very limited secure storage. A limitation of existing work is the predefined bound of the number of maximum ciphertext classes and key is prompt to leakage. Our proposed work mainly concentrates on above two problems. Our first work dynamically reserve number of maximum ciphertext classes in cloud storage. In case of Stream cipher the number of classes decided dynamically, because the cipher text size is too larger than block cipher. The number of classes required at dynamically is decided based in the prediction of previous delay of cipher sequences. Key stream depend only on the previous cipher texts and keys. This is asynchronous stream cipher whose i^{th} key is the function of n previous cipher texts. Additionally designing a Leakage-Resilient Identity-Based Encryption (LR-IBE) allows efficient and flexible key delegation. LR-IBE system is selectively secure under the simple Decisional Bilinear Diffie-Hellman assumption (DBDH), and serves as a stepping stone to our second fully secure construction. Our scheme gives an efficient key encryption scheme for efficient and flexible.

KEYWORDS*:* cloud storage, data sharing, asymmetric encryption, key- aggregate cryptosystem.

I. INTRODUCTION

Cloud storage is gaining popularity recently. In enterprise settings, we see the rise in demand for data outsourcing, which assists in the strategic management of corporate data. It is also used as a core technology behind many online services for personal applications. Now a days, it is easy to apply for free accounts for email, photo album, file sharing and/or remote access, with storage size more than 25 GB (or a few dollars for more than 1 TB). Together with the current wireless technology, users can access almost all of their files and emails by a mobile phone in any corner of the world. Considering data privacy, a traditional way to ensure it is to rely on the server to enforce the access control after authentication, which means any unexpected privilege escalation will expose all data. In a shared-ten an cycloud computing environment, things become even worse. Data from different clients can be hosted on separate virtual machines (VMs) but reside on a single physical machine. Data in a target VM could be stolen by instantiating another VM resident with the target one. Regarding availability of files, there are a series of cryptographic schemes which go as far as allowing a third-party auditor to check the availability of files on behalf of the data owner without leaking anything about the data, or without compromising the data owner's anonymity. Likewise, cloud users probably will not hold the strong belief that the cloud server is doing a good job in terms of confidentiality. A cryptographic solution, for example, with proven security relied on number-theoretic assumptions is more desirable, whenever the user is not perfectly happy with trusting the security of the VM or the honesty of the technical staff. These users are motivated to encrypt their data with their own keys before uploading them to the server.

ISSN (Online): 2319-8753
ISSN(Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 4, Special Issue 5, April 2015

International Conference On Emerging Trends in Engineering and Technology (ICETET'15)

On 13th & 14th March 2015

Organized by

Pandian Saraswathi Yadav Engineering College, Arasanoor, Sivagangai, Tamilnadu, India

II. EXISTING SYSTEM

Encryption keys also come with two flavors—symmetric key or asymmetric (public) key. Using symmetric encryption, when Alice wants the data to be originated from a third party, she has to give the encryptor her secret key; obviously, this is not always desirable. By contrast, the encryption key and decryption key are different in publickey encryption. The use of public-key encryption gives more flexibility for our applications. For example, in enterprise settings, every employee can upload encrypted data on the cloud storage server without the knowledge of the company's master-secret key.

Introducing a special type of public-key encryption which we call key-aggregate cryptosystem (KAC). In KAC, users encrypt a message not only under a public-key, but also under an identifier of ciphertext called class. That means the ciphertexts are further categorized into different classes. The key owner holds a master-secret called master-secret key, which can be used to extract secret keys for different classes. More importantly, the extracted key have can be an aggregate key which is as compact as a secret key for a single class, but aggregates the power of many such keys, i.e., the decryption power for any subset of ciphertext classes.

The sizes of ciphertext, public-key, master-secret key, and aggregate key in KAC schemes are all of constant size. The public system parameter has size linear in the number of ciphertext classes, but only a small part of it is needed each time and it can be fetched on demand from large (but nonconfidential) cloud storage.

Issues

- This work is the predefined bound of the number of maximum ciphertext classes.
- When one carries the delegated keys around in a mobile device without using special trusted hardware, the key is prompt to leakage.

III. PROBLEM DEFINITION

In cloud if data owner want to share his data or image files to user to securely send is possible in two ways 1. Data owner encrypts all files with a single encryption key and gives user the corresponding secret key directly.2. Data owner encrypts files with distinct keys and sends user the corresponding secret keys. Obviously, the first method is inadequate since all unchosen data may be also leaked to user For the second method, there are practical concerns on efficiency. The number of such keys is as many as the number of the shared files,say, a thousand. Transferring these secret keys inherently requires a secure channel, and storing these keys requires rather expensive secure storage. The costs and complexities involved generally increase with the number of the decryption keys to be shared. In short, it is very heavy and costly to do that. The best solution for the above problem is required for sharing data in cloud computing.

IV. PROPOSED SYSTEM

In proposed system we improve the Key-Aggregate Cryptosystem by determining the number of cipher text classes dynamically and applying ID based cryptosystem under Decisional Bilinear Diffie-Hellman assumption to protect leak information in cryptosystem which is called leak resilient cryptosystem. In the first work the classes of cipher text is decide based on the feedback of the current cipher delivered.

In this work a well suited software implementations is carried out to produce sequences of large period; thee repetitions will takes place after a large number of times and it should also have good statistical properties. It will take up this issue in subsequent class on pseudo randomness. So the proposed system number of classes is determined based on the above statistical properties.

In the second proposal we use the hash proof system that build a leakage-resilient IBE (LR-IBE) .In proposed system we use identity based encryption with Decisional Bilinear Diffie-Hellman assumption (DBDH). In this method for each identity id, there are many valid secret keys skid and also two kinds of ciphertexts: valid and invalid. The leakage resilient under the decisional bilinear Diffie-Hellman assumption (DBDH). The idea is to add another degree of randomness to our identity-based secret keys, called the "tag" t, coupled with some master secret key terms. This is done in a way that the secret-key holder can now only re-randomize the key along the original degree of freedom (which is needed for the original proof), but cannot re-randomize the key along the new "tag-dimension" anymore. This

ISSN (Online): 2319-8753
ISSN(Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization *Volume 4, Special Issue 5, April 2015*

International Conference On Emerging Trends in Engineering and Technology (ICETET'15)

On 13th & 14th March 2015

Organized by

Pandian Saraswathi Yadav Engineering College, Arasanoor, Sivagangai, Tamilnadu, India

will let us define invalid ciphertexts which decrypt to random values when the tag t is random, and yet decrypt to the same value when the tag t is kept the same, but the key is re-randomized along the original degree of freedom.

Advantages

- To protect against weak key-leakage attacks.
- The number of ciphertexts classes reserve dynamically.
- Efficient and flexible for key delegation.

V. PERFORMANCE METRICS

Delegation ratio and tree height are input parameter
Delegation ratio is the ratio of the delegated ciphertext classes to the total classes
For different delegation ratio and tree height the following output parameter is evaluated for base and proposed work

1. Compression Ratio
   This is the measurement how effectively save the costly secure storage requirement.
2. Number of granted keys
   Number of keys generated for data accessibility
3. Data Integrity (%)
   Data integrity refers to the validity of data. Data integrity checks the security bugs or threats and error occurrence when data is migrated from data owner to cloud server and cloud server to user
4. Confidentiality (%)
   The confidentiality of a system is ensured by discarding unauthorized access of segments in cloud storage.

METHOD

The key-aggregate cryptosystem (KAC). In KAC, users encrypt a message not only under a public-key, but also under an identifier of ciphertext called class. That means the ciphertexts are further categorized into different classes. The key owner holds a master-secret called master-secret key, which can be used to extract secret keys for different classes. More importantly, the extracted key have can be an aggregate key which is as compact as a secret key for a single class, but aggregates the power of many such keys, i.e., the decryption power for any subset of ciphertext classes.

With our proposed method, Data owner can simply send user a single aggregate key via a secure e-mail. User can download the encrypted files from downer's Dropbox space and then use this aggregate key to decrypt these encrypted files.

The sizes of ciphertext , public-key, master-secret key, and aggregate key in our KAC schemes are all of constant size. The public system parameter has size linear in the number of ciphertext classes, but only a small part of it is needed each time and it can be fetched on demand from large cloud storage.

VI. FUTURE IDEAS

A. **Setup**:
Executed by the data owner to setup an account on an untrusted server. On input a security level parameter 1_ and the number of cipher text classes n (i.e., class index should be an integer bounded by 1 and n), it outputs the public system parameter, which is omitted from the input of the other algorithms for brevity.

B. . **Key Generation**:
Executed by the data owner to randomly generate a public/master-secret key pair.

C. **Encrypt**:
Executed by anyone who wants to encrypt data. On input a public-key pk, an index I denoting the cipher text class, and a message m, it outputs a cipher text C.

D. **Extract**:
Executed by the data owner for delegating the decrypting power for a certain set of cipher text classes to a delegate. On input the master-secret key msk and a set S of indices corresponding to different classes, it outputs the aggregate key for set S denoted by KS.

E. **Decrypt**:

ISSN (Online): 2319-8753
ISSN(Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 4, Special Issue 5, April 2015

International Conference On Emerging Trends in Engineering and Technology (ICETET'15)

On 13th & 14th March 2015

Organized by

Pandian Saraswathi Yadav Engineering College, Arasanoor, Sivagangai, Tamilnadu, India

Executed by a delegate who received an aggregate key KS generated by Extract.
On input KS, the set S, an index i denoting the cipher text class the cipher text C belongs to, and C, it outputs the decrypted result m.

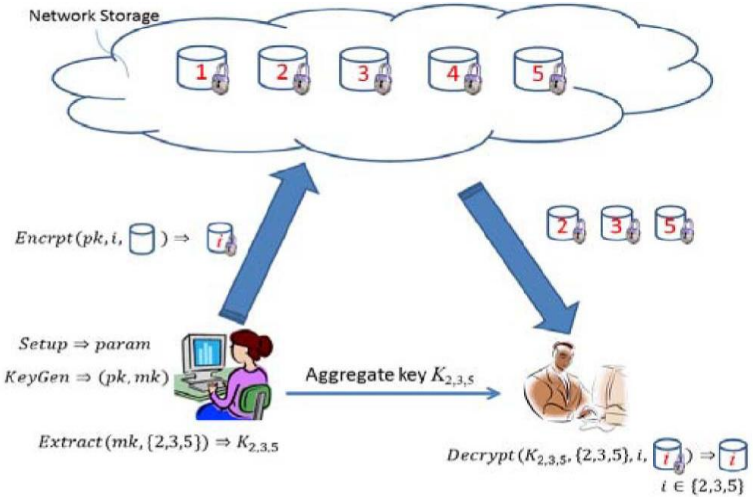

Fig. Using KAC for data sharing in cloud storage.

VI. MATHEMATICAL NOTATIONS

p,q, -prime numbers
1^{λ} -Input a security level
n -the number of ciphertext classes n
pk- public-key
i - An index denoting the ciphertext class
m -message,
C - ciphertext
msk -master-secret key
S -A set of indices corresponding to different classes
K_S - aggregate key for set S
Setup(1^{λ}, n) - executed by the data owner to setup an account on an untrusted server
KeyGen- data owner to randomly generate a public/master-secret key pair (pk, msk)
Encrypt(pk, I,m)- executed by anyone who wants to encrypt data
Extract(msk, S)- executed by the data owner for delegating the decrypting power for a certain set of ciphertext classes to a delegatee.
Decrypt(KS, S, I,C)- executed by a delegatee who received an aggregate key KS generated by Extract.

VII. CONCLUSION

We consider how to "compress" secret keys in public-key cryptosystems which support delegation of secret keys for different ciphertext classes in cloud storage. No matter which one among the power set of classes, the delegate can always get an aggregate key of constant size. Our approach is more flexible than hierarchical key assignment which can only save spaces if all key-holders share a similar set of privileges. The Leakage-Resilient Identity-Based Encryption (LR-IBE) systems from static assumptions in the standard model. We derive these schemes by applying a hash proof technique for Boneh-Boyen scheme. As a result, we achieve leakage-resilience under the respective static assumptions of the original systems in the standard model, while also preserving the efficiency of the original schemes.

ISSN (Online): 2319-8753
ISSN(Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization *Volume 4, Special Issue 5, April 2015*

International Conference On Emerging Trends in Engineering and Technology (ICETET'15)

On 13th & 14th March 2015

Organized by

Pandian Saraswathi Yadav Engineering College, Arasanoor, Sivagangai, Tamilnadu, India

REFERANCES

[1]Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage Cheng-Kang Chu, Sherman S.M. Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H. Deng, Senior Member, IEEE.
[2] S.S.M. Chow, Y.J. He, L.C.K. Hui, and S.-M. Yiu, "SPICE – Simple Privacy-Preserving Identity-Management for Cloud Environment," Proc. 10th Int'l Conf. Applied Cryptography and Network Security (ACNS), vol. 7341, pp. 526-543, 2012.
[3] C. Wang, S.S.M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy- Preserving Public Auditing for Secure Cloud Storage," IEEE Trans. Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.
[4] B. Wang, S.S.M. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," Proc. IEEE 33rd Int'l Conf. Distributed Computing Systems (ICDCS), 2013.
[5] S.S.M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R.H. Deng, "Dynamic Secure Cloud Storage with Provenance," Cryptography and Security, pp. 442-464, Springer, 2012.
[6] G. Ateniese, 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.