ISSN (Online) : 2454-4159

IJARSMT International Journal of Advanced Research in Science Management and Technology

Volume 2, Issue 7, July 2016

Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage

Shubham Patil[1],Prof. Kailash Patidar[2] , Manoj Yadav[3], Rishi Kushwah[4]

PG Student, Dept. of CSE, SSSUTMS, Sehore, M.P., India[1]
Professor & Head, Dept. of CSE, SSSUTMS, Sehore, M.P., India[2]
Assistant Professor, Dept. of CSE, SSSUTMS, Sehore, M.P., India[3]
Assistant Professor, Dept. of CSE, SSSUTMS, Sehore, M.P., India[4]

ABSTRACT- "Data sharing is an important functionality in cloud storage. In this article, we show how to securely, efficiently, and flexibly share data with others in cloud storage. We describe new public-key cryptosystems which produce constant-size cipher texts such that efficient delegation of decryption rights for any set of cipher texts are possible. The novelty is that one can aggregate any set of secret keys and make them as compact as a single key, but encompassing the power of all the keys being aggregated. In other words, the secret key holder can release a constant-size aggregate key for flexible choices of cipher text set in cloud storage, but the other encrypted files outside the set remain confidential. This compact aggregate key can be conveniently sent to others or be stored in a smart card with very limited secure storage. We provide formal security analysis of our schemes in the standard model. We also describe other application of our schemes. In particular, our schemes give the first public-key patient-controlled encryption for flexible hierarchy, which was yet to be known.

KEYWORDS- Cloud storage, public key encryption, cryptosystem, key aggregate encryption, and 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. Nowadays, it is easy to apply for free accounts for email, photo album, file sharing and/or remote access, with storage size more than 25GB (or a few dollars for more than 1TB). 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 unexpected privilege escalation will expose all data. In a shared-tenancy cloud 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 co-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 ownersanonymity. 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, 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. Data sharing is an important functionality in cloud storage

II. LITERATURE SURVEY

The project consists of four interrelated thrusts: High availability is one of the key characteristics of Infrastructure-as-a-Service (IaaS) cloud. In this paper, we show a scalable method for availability analysis of large scale IaaS cloud using analytic models. To reduce the complexity of analysis and the solution time, we use an interacting Markov chain based approach. The construction and the solution of the Markov chains is facilitated by the use of a high-level Petri net based paradigm known as stochastic reward net (SRN). Overall solution is composed by iteration over individual SRN sub-model solutions. Dependencies among the sub-models are resolved using fixed-point iteration, for which existence of a solution is proved. We compare the solution obtained from the interacting sub-models with a monolithic model and show that errors introduced by decomposition are insignificant. Additionally, we provide closed form solutions of the sub models and show that our approach can handle very large size IaaS clouds. Cloud computing is a model of Internet-based computing. An IaaS cloud, such as Amazon EC2 and IBM Smart Business Cloud delivers, on-demand, operating system (OS) instances provisioning computational resources in the form of virtual machines deployed in the cloud provider's data center. Requests submitted by the users are provisioned and served if the cloud has enough available capacity in terms of physical machines. Large cloud service providers such as IBM provide service level agreements (SLAs) regulating the availability of the cloud service. Before committing an SLA to the customers of a cloud, the service provider needs to carry out availability analysis of the infrastructure on which the cloud service is hosted. In this paper, we show how stochastic analytic models can be utilized for cloud service availability analysis. We first develop a one-level monolithic model. However, such monolithic models become intractable as the size of cloud increases. To overcome this difficulty, we use an interacting sub-models approach. Overall model solution is obtained by iteration over individual sub-model solutions. Comparison of the results with monolithic model shows that errors introduced by model decomposition are negligible. We also develop closed form solutions of the sub-models and show that our approach can scale for large size clouds. To the best of our knowledge, this is the first attempt to analyze availability of a cloud computing infrastructure by using stochastic analytic models. The presence of three pools of physical machines and the migration of them from one pool to another caused by failure events makes the model both novel and interesting. In order to automate the construction and solution of underlying Markov models, we use a variant of stochastic Petri net (SPN) called stochastic reward net (SRN).

In daily life generation cloud computing is visualizes architecture. The risk of attacker can attack the private data on user and leak the user identity. While the need is authentication is necessary for cloud user and service provider. The service provider and users both are not compromised then issue will be arises. The need of user is Accessing data locally that present on remote side with flexible use of cloud storage. There is need to inspect the data set on cloud. There is many cloud users that wants to upload data without using its personal information. The Attribute-Based Encryption (ABE) is another way to sharing encrypted data. Encrypting each part of data is worse than equivalent user attribute to encrypt data. The attribute decrypt the cipher text is matched only a particular key in Attribute-Based Encryption. For Decrypting a particular Cipher text the user key and attribute must be match. A hierarchical access control accomplished by Multi group key management by applying multiple group authorities with handling group keys for different user and also integrated key graph. Tree structure is used by centralized key management plan to minimize Storage overload, Communication and Processing of Data. It updates and also maintains keying related things. For every users an integrated key graph is accomplished. A vital primary thing of Identity based Cryptography is Identity-Based Encryption. Different information of user's identity is contained by the public key of user. The Domain name or textual value can be Key. The public Key infrastructure is deployed using IDE. For public key encryption identity of the user is used as identity string .In IBE another name of trusted party is private key generator which gives according to user identity master secrete key and secrete key. The identity of user to encrypt data and public value collaborated by data owner. By using secrete key decrypt the cipher text. The user must get a particular key related to attribute while decrypting a message in multi attribute authorities number of attribute are analyzed regarding the decryption key. The user those who have attribute identity without interaction between each other are allocated independently decryption keys. Real time deployment attributes which is based on privileges is allowed by multi authority attribute based encryption as different attribute are issued by different authority. Confidentiality is maintain by central authority due to attribute authority ensure the honesty of user privileges.

III. PROBLEM DEFINITION

The proposed system problem statement is "To design adaptable and efficacious public key encryption that will delegate any subset of cipher texts (produced by encryption scheme) and is decrypt able by a fixed - size decryption key."

Therefore, the above design is a special type of public-key encryption which we call Key-Aggregate Cryptography (KAC).

- **PROBLEM IN EXISTING SYSTEM**

1. The costs and complexities involved generally increase with the number of the decryption keys to be shared.
2. The encryption key and decryption key are different in public key encryption.

- **SOLUTION OF THESE PROBLEMS**

The above system has some drawbacks

1. The extracted key have can be an aggregate key which is as compact as a secret key for a single class.
2. The delegation of decryption can be efficiently implemented with the aggregate key.

IV. PROPOSED SYSTEM

A Key Aggregate Encryption technique consist of five polynomial time algorithm which given below. This polynomial time algorithm which is used for encryption of data. The data owner which generate setup parameter using setup and generate the master secrete key which is used for encrypt the data using key gen. Encrypt function is used for encryption of message also it defines cipher text classes.

Extract function is used for extracting the secrete key to generate the aggregate key and this aggregate key is used for decryption of data. After that Decrypt function is used. This Decrypt function uses the aggregate key for decrypting the data.

A.Setup Phase

Using setup phase data owner can create account on untrusted server. This setup phase takes only the security parameters which are used for performing setup operation.

B.KeyGen Phase

The data owner is execute the keygen phase for generating the private key master secrete key (pk, msk).

C.Encrypt Phase

Anyone can execute encrypt phase who wants to store encrypted data on server. Encrypt function takes the actual original massage, private key and index (pk,m,i). The encryption algorithm takes the input message m and produces output cipher text c. Only the users who has set of attributes which are used for decryption.

D.Extract Phase

The extraction phase is executed by the data owner, who gives the delegating power to the delegate for the certain set of cipher text classes.

Decrypt Phase

The decryption phase is executed by delegate after receiving the aggregate key which is used for decryption of data (Ks, S, i, c).

We considered the verifiability of the cloud's transformation and provided a method to check the correctness of the transformation. However, the we did not formally define verifiability. But it is not feasible to construct ABE schemes with verifiable outsourced decryption following the model defined in the existing. Moreover, the method proposed in existing relies on random oracles (RO). Unfortunately, the RO model is heuristic, and a proof of security in the RO model does not directly imply anything about the security of an ABE scheme in the real world. It is well known that there exist cryptographic schemes which are secure in the RO model but are inherently insecure when the RO is instantiated with any real hash function.

In this thesis work, firstly modify the original model of ABE with outsourced decryption in the existing to allow for verifiability of the transformations. After describing the formal definition of verifiability, we propose a new ABE model and based on this new model construct a concrete ABE scheme with verifiable outsourced decryption. Our scheme does not rely on random oracles.

In this paper we only focus on CP-ABE with verifiable outsourced decryption. The same approach applies to KP-ABE with verifiable outsourced decryption.To assess the performance of our ABE scheme with verifiable outsourced decryption, we implement the CP-ABE scheme with verifiable outsourced decryption and conduct experiments on both an ARM-based mobile device and an Intel-core personal computer to model a mobile user and a proxy, respectively.

Advantages of Proposed System:

1. The extracted key have can be an aggregate key which is as compact as a secret key for a single class.
2. The delegation of decryption can be efficiently implemented with the aggregate key.

V. CONCLUSION AND FUTURE WORK

How to protect users' data privacy is a central question of cloud storage. With more mathematical tools, cryptographic schemes are getting more versatile and often involve multiple keys for a single application. In this article, 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 delegatee 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. A limitation in our work is the predefined bound of the number of maximum ciphertext classes. In cloud storage, the number of cipher texts usually grows rapidly. So we have to reserve enough ciphertext classes for the future extension. Otherwise, we need to expand the public-key as we describe. Although the parameter can be downloaded with cipher texts ,it would be better if its size is independent of the maximum number of ciphertext classes. On the other hand, when one carries the delegated keys around in a mobile device without using special trusted hardware, the key is prompt to leakage, designing a leakage resilient cryptosystem yet allows efficient and flexible key delegation is also an interesting direction. Developer productivity and application quality and specifying maintenance. You might never want to write another line of SQL again.

REFERENCES

[1] S. S. M. Chow, Y. J. He, L. C. K. Hui, and S.-M.Yiu, “SPICE Simple Privacy-Preserving Identity-Management for Cloud Environment,” in Applied Cryptography and Network Security - ACNS 2012, ser. LNCS, vol. 7341. Springer, 2012, pp. 526–543.

[2] L. Hardesty, “Secure computers aren’t so secure,” MIT press, 2009, http://www.physorg.com/news176107396.html.

[3] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, “PrivacyPreservingPublicAuditingforSecureCloudStorage,”IEEETrans. Computers, vol. 62, no. 2, pp. 362–375, 2013.

[4] B. Wang, S. S. M. Chow, M. Li, and H. Li, “Storing Shared Data on the Cloud via Security-Mediator,” in International Conference on Distributed Computing Systems - ICDCS 2013. IEEE, 2013.

[5] S. S. M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R. H. Deng, “Dynamic Secure Cloud Storage with Provenance,” in Cryptography and Security: From Theory to Applications - Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday, ser. LNCS, vol. 6805. Springer, 2012, pp. 442–464.

[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, “Aggregate and Verifiably Encrypted Signatures from Bilinear Maps,” in ProceedingsofAdvancesinCryptology-EUROCRYPT’03,ser.LNCS, vol. 2656. Springer, 2003, pp. 416–432.

[7] M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, “Dynamic and Efficient Key Management for Access Hierarchies,” ACM Transactions on Information and System Security (TISSEC), vol. 12, no. 3, 2009.