Volume 5, Issue 11, November 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Resource Sharing Using Key Aggregation and Information Flow Control (RAIFC)

Ranjitha Chandramouli, Dipita Agrawal, Pallavi Dod, Pranav Kulkarni, Prof Priyanka Kedar

Computer Department, Dhole Patil College, Pune, Maharashtra,

India

Abstract: Information flow control (Ifc) is used to avoid unauthorized access of information. Data sharing is an important functionality in distributed system. In this proposal we create distributed system in which user can open its account using Aadhar card and update their documents which user want and then they are encrypted. If user is facing any problem such as insufficient documents or lack of quality of documents at bank or reservation sites, then company or agency which wants documents of customer will request for accessing. Specific documents will be downloaded from this government site by agencies directly. For such a site security is an important an issue or else these documents may be misused, under such condition aggregate key concept is used. If anyone wants to access documents of customer then indirectly key is generated and this key is used as password for accessing particular documents. This aggregate key is sent to customer and if verified then this key is transferred to the agency. Only then by using this key agency or company gets an authority to access those documents. So there is no need to revisit there and it makes customers work easier.

Keywords: Key Aggregation, Distributed System, Data Sharing

I. INTRODUCTION

In this project, we are going to make the resource sharing secure in the Distributed System by using key aggregation and information flow control(IFC). [2]It will make the work of user much easier. In this proposal, we show how to securely, efficiently, and flexibly share data with others in distributed system. In this proposal we create distributed system in which user can open its account using Aadhar Card and update their documents which user wants. And these documents are updated on government site. Then these documents are verified from specific government agencies and then they are encrypted. Status of documents is seen that means the documents are verified or not. Specific documents will be downloaded from this government site by agencies directly. For such a site security is an important an issue or else these documents may be misused, under such condition aggregate key concept is used. This key depends on number of documents. The other encrypted files outside the set remain confidential. This aggregate key is sent to customer and if verified then this key is transferred to the agency. Only then by using this key agency or company gets an authority to access those documents. This system provides paperless documentation process which is environment friendly.

II. KEY-AGGREGATE CRYPTOSYSTEM

In key-aggregate cryptosystem (KAC), users encrypt a messagenot only under a public-key, but also under an identifier of ciphertext called class. That means the ciphertexts arefurther 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.[1]With our example, Alice can send Bob a single aggregate key through a secure e-mail. Bob can download the encrypted photos from Alice's Box.com space and then use this aggregate key to decrypt these encrypted data.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 non-confidential) distributed storage.

III. EXISTING SYSTEM

Key management and key sharing plays an important role in data sharing of the distributed systems. Existing system is not secure and efficient. The costs and complexities involved in sharing of data increase with the number of decryption keys. In public key encryption technique, the encryption and decryprion key is different.Traditional key cryptosystems lack the security feature as they were generated by the random key generation techniques. In early days, processing machinery was bulky & expensive, so resources had to be shared to make them cost-effective.

© 2015, IJARCSSE All Rights Reserved

IV. PROPOSED SYSTEM

In this paper, we study how to make a decryption key more powerful in the sense that it allows decryption of multiple ciphertexts, without increasing its size.[5] Specifically, our problem statement is "To design an efficient public-key encryption scheme which supports flexible delegation in the sense that any subset of the ciphertexts (produced by the encryption scheme) is decry ptable by a constant-size decryption key (generated by the owner of the master-secret key)." The problem is solved by introducing a special type of public-key encryption which is called as the key aggregate cryptosystem (KAC). [6]

Key Aggregation technique is much more secure as the Decryption key is sent via a secure channel and is kept disclosed. The Aggregate key acts like a secret key A key aggregate encryption has five polynomial-time algorithms :

Setup Phase, KeyGen Phase, Encrypt Phase, Extract Phase,

Decrypt Phase.

1. Setup phase: The data owner executes the setup phase for an account on server which is not trusted. The setup algorithm only takes implicit security parameter
2. KeyGen Phase: This phase is executed by data owner to generate the public or the master key pair (pk, msk) .
3. Encrypt Phase: This phase is executed by anyone who wants to send the encrypted data.Encrypt (pk, m, i), the encryption algorithm takes input as public parameters pk, a message m, and i denoting ciphertext class. The algorithm encrypts message m and produces a ciphertext C such that only a user that has a set of attributes that satisfies the access structure is able to decrypt the message
   Input= public key pk, an index i, and message m
   Output = ciphertext C
4. Extract Phase: It is executed by data owner for delegating the decrypting power for a certain set of ciphertext classes and it outputs the aggregate key for set .
5. Decrypt Phase: This is executed by the candidate who has the decryption authorities. Decrypt (kS, S, i, C), the decryption algorithm takes input as public parameters pk, a ciphertext C, i denoting ciphertext classes for a set S of attributes.
   Input = kS and the set S, where index i = ciphertext class
   Outputs = m if i element of S

Advantage:

1. Improved data sharing.
   It helps create an environment in which end users have better access to more and better-managed data. Such access makes it possible for end users to respond quickly to changes in their environment.
2. Improved data security.
   The more users access the data, the greater the risks of data security breaches. Corporations invest considerable amounts of time, effort, and money to ensure that corporate data are used properly.
3. Better data integration.
   Wider access to well-managed data promotes an integrated view of the organization's operations and a clearer view of the big picture.
4. Minimized data inconsistency.
   Data inconsistency exists when different versions of the same data appear in different places.

Limitations:

1. Management complexity.
   Given the fact that database systems hold crucial company data that are accessed from multiple sources, security issues must be assessed constantly.
2. Maintaining currency:
   To maximize the efficiency of the database system, you must keep your system current.

V. CONCLUSION

Users data privacy is a central question of distributed storage. Data sharing is important concept. Compress secret keys in public-key cryptosystems which support delegation of secret keys for different cipher text classes in distributed system. No matter which one among the power set of classes, the delegate can always get an aggregate key of constant size. The data sharing concepts in distributed system. using key aggregation and information flow control is efficient and secure.

REFERENCES

[1] Cheng-Kang Chu ,Chow, S.S.M, Wen-Guey Tzeng, Jianying Zhou, and Robert H. Deng, ―Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage‖, IEEE Transactions on Parallel and

[2] F. Guo, Y. Mu, and Z. Chen, ―Identity-Based Encryption: How to Decrypt Multiple Ciphertexts Using a Single Decryption Key, in Proceedings of Pairing-Based Cryptography (Pairing '07), ser. LNCS, vol. 4575. Springer, 2007, pp. 392–406.

© 2015, IJARCSSE All Rights Reserved

[3] S. S. M. Chow, Y. Dodis, Y. Rouselakis, and B. Waters, ―Practical Leakage-Resilient Identity-Based Encryption from Simple Assumptions,‖ in ACM Conference on Computer and Communications Security, 2010, pp. 152–161.

[4] V. Goyal, O. Pandey, A. Sahai, and B. Waters, ―Attribute-Based Encryption for Fine-Grained Access Control of Encrypted data,‖in Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS ’06). ACM, 2006, pp. 89–98.

[5] M. Chase and S. S. M. Chow, ―Improving Privacy and Security in Multi-Authority Attribute-Based Encryption,‖ in ACM Conference on Computer and Communications Security, 2009, pp. 121 –130.

[6] T. Okamoto and K. Takashima, ―Achieving Short Ciphertexts or Short Secret-Keys for Adaptively Secure General Inner-Product Encryption,‖ in Cryptology and Network Security (CANS ’11), 2011, pp. 138–159.

© 2015, IJARCSSE All Rights Reserved