LIGHT WEIGHT HASH FUNCTION FOR SCALABLE DATA SHARING IN CLOUD STORAGE

[1]H.Hussainiya Fathima, [2]V.R. Sathish Kumar.
[1]PG Scholar, Department of Computer Science and Engineering, Mohamed Sathak Engineering College, Ramanathapuram, India.
[2]Assistant professor, Department of Computer Science and Engineering, Mohamed Sathak Engineering College, Ramanathapuram, India
[1]fathimahb10@gmail.com [2]sksathish725@gmail.com

Abstract: The data sharing in cloud needs security for the items which are shared to some others in the cloud. The concept of light weight hash function based data sharing in the cloud is used for secure sharing. This work easily maintains the scalability and throughput. The process behind the proposed system is to share the data from sender to receiver in the cloud. First, the users named as senders and receivers are entering into the cloud with the help of registration process. At the time of registration the secret key can be generated with the help of DSA algorithm. The key can be send to the user's mail. After getting the key, enter into the cloud by the way of login. Then the sender can choose a file to be send and it can be encrypted with light weight hash function. This produces, public key for encrypting and private key decrypting the files. The results of the experiments can increase latency of the network.

Index Terms: Light weight Hash Function; Latency; Secret key; Data Sharing.

1. INTRODUCTION

Cloud storage can providethe benefits of greater accessibility and reliability; rapid deployment; strong protection for backup and disaster recovery purposes; lowers the overall storage costs as a result of not having to purchase. Also known as mobile cloud storage, personal cloud storage is a descendant of public cloud storage that applies to storing an individual data in the cloud and providing the individual with access to the data from anywhere. provides data syncing and sharing capabilities across multiple devices.

The key aggregation property is especially useful when we expect the delegation to be highly efficient and flexible. Cryptographic key assignment schemes aim to lower the expense in storing and managing secret keys. Using a tree structure, a key for a given branch can be used to derive the keys of its descendant nodes (but not the other way round). Granting the parent key implicitly grants all the keys of its descendant nodes.

A method is used to generate a tree hierarchy of symmetric keys by using unlimited evaluations of pseudorandom function/block-cipher on a fixed secret. The concept can be more specialized from a tree to a graph. More programmed cryptographic key assignment schemes also supports access policy that can be modeled by an acyclic graph or a Cyclic graph. Most of these schemes produces keys for symmetric-key cryptosystems, despite the key derivations may require modular arithmetic as used in public-key cryptosystems.

2. RELATED WORKS

It is unrealistic to assume there is a single authority which can monitor every single attribute of all users. Multi-authority attribute-based encryption capacitate a more realistic deployment of attribute-based access control, such that different authorities are responsible for issuing different sets of attributes[1].

Since key material is the most important concern in unconditionally secure verification, given the message is encrypted with a perfect secret one-time pad cipher, achieve unconditionally secure authentication with almost free key material. Propose a method for unconditionally authenticating arbitrarily long messages with much shorter keys [2].

Key reduction is achieved by utilizing the special structure of the authenticated encryption.

3. EXISTING METHOD

A sanctioned application of KAC is data sharing. The key aggregation property is particularly beneficial when we expect the authorization to be adept and bendable. The schemes enable a content provider to share their data in a entrusted and particular way, with a fixed and small cipher text enlargement, by spreading to each

authorized user a single and small aggregate key.

Cryptographic key assignment schemes aim to reduce the expense in storing and conducting secret keys for general cryptographic use. Deploying a tree structure, a key for a given branch can be pre-owned to derive the keys of its descendant nodes. Just concede the parent key implicitly grants all the keys of its descendant nodes.

A method to generate a tree hierarchy of symmetric keys by using iterated estimations of pseudorandom function/block-cipher on a fixed secret. The concept can be unspecialized from a tree to a graph. More innovative cryptographic key assignment schemes support access policy that can be procedure by an acyclic graph or a cyclic graph.

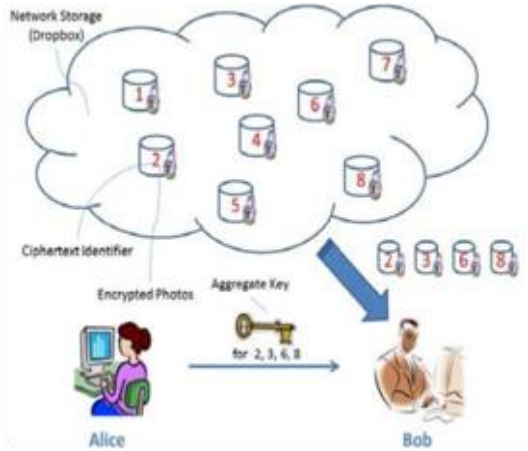

Figure 1: Sharing Files

Most of these schemes produce keys for symmetric-key cryptosystems, against the key Inferences may require commutable arithmetic as used in public-key cryptosystems.

A key-aggregate encryption scheme is composed of five polynomial-time algorithms. The data owner creates the public system parameter by Setup and generates a public/master-secret key pair by KeyGen. Messages can be encrypted by Encrypt That is, authentication exploits the privacy of the message to reduce the key material required for authentication. [5] by anyone who decides what cipher text class is related with the plaintext message to be encrypted.

The data owner handles the master-secret to generate an aggregate decryption key for a set of cipher text Classes Extract. The generated keys can be passed to representative securely. Any user with an aggregate key can decrypt any cipher text presuming that the cipher text's class is enclosed in the aggregate key via Decrypt.

4. PROPOSED METHOD

In proposed system the users must register into the cloud system. At the time of registration they can provide some information related to them particularly unique name and id. After the successful completion of the registration the digital signature can be generated to user's mail. The signature can be accomplished with the help of DSA algorithm. After the registration user can login to the cloud. At that time the user must give their individual signature because according to that signature the validation of the user can be made.

If the user can give any invalid signature means they will not enter into the system. After entering the cloud user need to perform the operations of file sharing. Here, the file can be encrypted with the help of light weight hash function. This generates the public key and also the private key. The sender can send the public key to the receiver side. The receivers decrypt the notification with the private key.

4.1 Authentication

First the user can enroll their details into cloud. After registration the digital signature key can be send to user's mail. According to the key the user can login to the cloud system and do further process. For digital signature process we have to use DSA algorithm. DSA algorithm is used to generate a secret key for each user in the cloud.

This system maintains the scalability and throughput. The process behind is to share the data's from sender to receiver. Initially, the users enters into the cloud with the help of registration . At the time of registration the secret key can be generated with of DSA algorithm.

The key can be send to the user's mail. After getting the key user have to enter into the cloud the way of login. Then the sender can choose a file to be sent and that can be encrypted with key
aggregate process. For such purpose, use light weight hash function. This produces the same public key for decrypting the files. The same process will be made at the receiver end of the cloud. The results of the experiments can increase latency of the network.

.

4.2 File Transfer

The user can select the images from the system and for sending to another user in the receiver side. The total selected images are to be 10 in the folder. But another user wants only 3 images from the folder, other images are not known to another user. For that purpose we have to use the encryption technique for the folder of images. Functions with these properties are used as

hash functions for a variety of intent. Practical applications includes message integrity checks, digital signature, authentication, and various information of security applications hash function takes a string of any length as input and produces a fixed length string which acts as a kind of "signature" for the data provided.

In this way, a person knowing the "hash value" is unable to know the authentic message, but only the person who knows the authentic message can prove the "hash value" is created from that message. A cryptographic hash function should behave as much as possible like a random function while still being deterministic and efficient computable. A cryptographic hash function is considered "insecure" from a cryptographic point of view.

Design a new system that contains a limited no of keys to encrypt and decrypt operation. Mobile user can easily send a data to the receiver with a secure key and maintain the confidential. In our designed system must improve the reliability of cloud system by using simple light weight hash function.

The sender, searches the requested images of the receiver and compresses all the images in a single folder and a single public key is generated for the compressed folder, and forwarded to the receiver, by mail. When, the receiver tries to decrypt the images, verification of digital signature is carried out

4.3 Light Weight Hash Function Encryption

The sender can use the light weight hash function for encrypting that files. The function of light weight hash function is to encrypt the files it could generate the public key and also private key. Here, the send can use the public key for sending process. At the end of receiver side the private key can be used to decrypt the files. Data clustering is an effective method for data analysis and pattern recognition which has been applied in many fields such as image segmentation.

It is a process of classifying the multidimensional data into several groups or clusters based on some similarity measures. A cluster is usually defined by a cluster center. The information of the features may differ from each other and the contribution to the clustering are different.

Some features play an important role in explaining the differences among the samples, thus should pay a more attention in the clustering process to get a exact grouping. In order to emulate the particular contribution of the feature, this paper proposes a new feature weighted affinity propagation clustering (AP) algorithm. LHash is based on extended sponge functions framework, which allows trade-offs among archetypal, speed, energy consumption and implementation cost by adjusting parameters.

The internal permutation is designed using a structure, named as Feistel-PG, which is an extended variant of improved generalized Feistel. Feistel-PG has fast diffusion, shorter differential paths and integral distinguishers than similar structures. The S-boxes and MDS linear layer used in the internal permutation are designed to be hardware-friendly. Both have very compact hardware development. The MDS linear layer has an iterated implementation, which is similar to and even more compact than the linear layer used in photon.

We present the LHash achieves remarkably compact implementation in hardware. In our smallest implementation, the area requirements are 817 and 1028 GE with 666 and 882 cycles 5 SYSTEM FLOW DIAGRAM per block, respectively. Meanwhile, it is efficiency on energy consumption evaluated by the metric of energy per bit proposed in is the smallest class among current lightweight hash functions in literature.Especially, for the competitors with similar pre image and collision resistance levels, it also competes well in terms of area and throughput tradeoff.

4.4 Decryption Process

The decryption process is to be done in the receiver side. For decryption process the user can use the light weight hash function. This will provide the secured decryption in a system. The receiver wants particular images from the folder means they get the private key of those images from sender side with the help of sending requests

5. SYSTEM FLOW DIAGRAM

The overall system design reviews, that the user can enroll their details into cloud. After registration the digital signature key can be send to user's mail. According to the key the user can login to the cloud system and do further process. For digital signature process we have to use DSA algorithm.

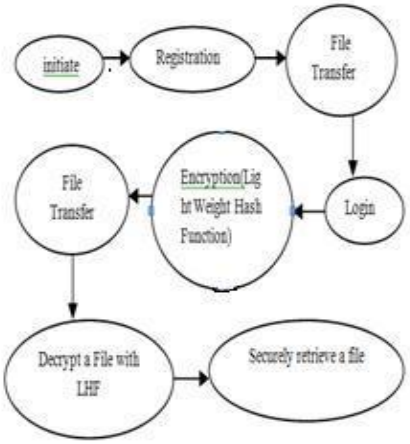

Figure 2: system flow diagram

The user can select the images from the system and for sending to another user in the receiver side. Then encrypt the files which can be sending by the sender here the sender can use the light weight hash function for encrypting that files. The function of light weight hash function is to encrypt the files it could generate the public key and also private key. Here, the send can use the public key for sending process. At the end of receiver side the private key can be used to decrypt the files

6. EXPERIMENTAL RESULTS

To protect user's data privacy is a central question of cloud storage. We consider how to "Compress" Secret keys in public key cryptosystems which support delegation of secret keys for different cipher text classes in cloud storage. No concern which one among the power set of associated classes, the delegate can always get an addregate key of constant size. Our Approach 2 than hierarchical key assignment which can only save spaces if all key-holders share a similar set of privileges. A light weight hash function takes a string of any length as input and produce a fixed length string which acts as a kind of "Signature". Initially registration takes place and once the registration is completed, a secret key is generated by DSA algorithm. Both the process of encryption and decryption is carried out, only with the verification of secret key that has been generated. Fig (a) shows the process of registration. Fig (b) shows the process of light weight hash encryption where, the images can be encrypted using light weight hash function.

(a)

(b)

(c)

(d)

Figure 3: Results of the proposed method: (a) Registration process; (b) Light Weight Hash function applied for the purpose of encryption; (c) choosing the number of images using affinity propagation method ;(d) secure way of decrypting the images

7. CONCLUSION AND FUTURE WORK

7.1 Conclusion

Overcomes the drawbacks associated with the existing system. In proposed concept, provide a secure sharing of data in cloud environment. Security is achieved through DSA algorithm.

7.2 Future work

In future, this can be expanded by enhancing another segment of algorithm for improvising the security. Also enhance to share the videos between the sender and the receiver. Try to develop the algorithm more simplified for the smart phone users.

8. ACKNOWLEDGEMENT

The author would like to thank the respective Head of the Institution, Head of the Department and Faculty members for giving valuable advices and providing technical support.

REFERENCES

[1] T.H. Yuen, S.S.M. Chow, Y. Zhang, and S.M. Yiu,"Identity-Based Encryption Resilient to Continual Auxiliary Leakage," Proc. Advances in Cryptology Conf. (EUROCRYPT '12), vol. 7237, pp. 117-134, 2012.

[2] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical Identity Based Encryption with Constant Size Ciphertext," Proc. Advances in Cryptology Conf. (EUROCRYPT '05), vol. 3494, pp. 440-456, 2005.

[3] D. Boneh, R. Canetti, S. Halevi, and J. Katz, "Chosen-Ciphertext Security from Identity-Based Encryption," SIAM J. Computing, vol. 36, no. 5, pp. 1301-1328, 2007.

[4] R. Canetti and S. Hohenberger, "Chosen-Ciphertext Secure Proxy Re-Encryption," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 185-194, 2007.

[5] C.-K. Chu and W.-G. Tzeng, "Identity-Based Proxy Re-encryption without Random Oracles," Proc. Information Security Conf. (ISC '07), vol. 4779, pp. 189-202, 2007.

[6] C.-K. Chu, J. Weng, S.S.M. Chow, J. Zhou, and R.H. Deng, "Conditional Proxy Broadcast Re-Encryption," Proc. 14th Australasian Conf. Information Security and Privacy (ACISP '09), vol. 5594, pp. 327-342, 2009.

[7] S.S.M. Chow, J. Weng, Y. Yang, and R.H. Deng, "Efficient Unidirectional Proxy Re-Encryption," Proc. Progress in Cryptology (AFRICACRYPT '10), vol. 6055, pp. 316-332, 2010.

[8] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," ACM Trans. Information and System Security,vol. 9, no. 1, pp. 1-30, 2006.

[9] D. Boneh, C. Gentry, and B. Waters, "Collusion Resistant BroadcastEncryption with Short Ciphertexts and Private Keys," Proc.Advances in Cryptology Conf. (CRYPTO '05), vol. 3621, pp. 258-275,2005.

[10] L.B. Oliveira, D. Aranha, E. Morais, F. Daguano, J. Lopez, and R. Dahab, "TinyTate: Computing the Tate Pairing in Resource-Constrained Sensor Nodes," Proc. IEEE Sixth Int'l Symp. NetworkComputing and Applications (NCA '07), pp.318-323, 2007.

[11] T. Okamoto and K. Takashima, "Achieving Short Ciphertexts or Short Secret-Keys for Adaptively Secure General Inner-Product Encryption," Proc. 10th Int'l Conf. Cryptology and Network Security (CANS '11), pp. 138-159, 2011.