KEY-AGGREGATE CRYPTOSYSTEM FOR SHARING THE DATA

Shanmugapriya B[1] Rakavi S[2] Revathi v[3] Sujatha.V[4]

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, KCG COLLEGE OF TECHNOLOGY, CHENNAI

Priyabalasundaram1@gmail.com1 ammuraj257@gmail.com2 revathikanna123@gmail.com 3 sujatha@kcgcollege.com4

Abstract-. Goal of the project is to provides an secure environment to share data and files over the network. An user normally can share the files to their colleagues and restrict the access of another group but, a person in their group can download the work of a another person and overwrite it and it provides an single key for both encryption and decryption. This project provides an secured environment to share the files within the particular group, the user can achieve an effective way for data sharing among the group, it provides multiple keys for each user in a particular group to share files .It performs multiple encryption over the shared files and restrict the access to download files within the same group and it provides separate keys for encryption and decryption.

This project restrict the access to download files of a another colleagues

and thus provides an secured environment to share files over the network.

INTRODUCTION

The main aim of our project is to sharing the data in cloud. In this paper, we show that how to securely, efficiently and flexibly share data others in cloud storage. For that we propose Key-Aggregate Cryptosystem which produces cipher text of constant size such that decryption rights can be assigned on them. By combining a set of secret key, we can make a compact single key. By using this compact key, we can send others or can be store in a very limited secure storage. First, owner of the data Setup the public system next KeyGen algorithm generates a public or master/secret key. By using this key, user can convert plain text to cipher text. Next user will give input as master secret key by Extract function; it will produce output as aggregate decryption key. This generated key is safely sent to the receiver. Then the user with aggregate key can decrypt the cipher text through the use of Decrypt function. 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. 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.

The existing system of cloud storage bloggers can let their friends view a subset of their private pictures or data; an enterprise may grant her employees access to a portion of sensitive data. The challenging problem is how to effectively share encrypted data. Of course users can download the encrypted data from the storage, decrypt them, then send them to others for sharing, but it loses the value of cloud storage. Users should be able to delegate the access rights of the sharing data to others so that they can access these data from the server directly. However, finding an efficient and

secure way to share partial data in cloud storage is not trivial. The receiver decrypting the original Message using symmetric key algorithm.

RELATED WORK

Setup Public System

The setup algorithm takes no input other than the implicit security parameter. It outputs the public parameters generate the public key PK and a master key MK. They generate the number of ciphertext class and index also.

Key Generation

Key Generation (MK, S). The key generation algorithm takes as input the master key MK and a set of attributes S that describe the key. It outputs a private key sender. the data owner to randomly generate a public/master-secret key .

Key Aggregate Encryption

The Modern cryptographic systems include AES algorithm. AES algorithm uses a single shared key; keeping data secret requires keeping this key secret. Public-key algorithms use a public key and a private key. The public key is made available to anyone (often by means of a digital certificate). A sender encrypts data with the public key; only the holder of the private key can decrypt this data.

The encryption algorithm takes as input the public parameters PK, a message M, and an access structure A over the universe of attributes. The algorithm will encrypt M and produce a cipher text CT such that only a user that possesses a set of attributes that satisfied the access structure will be able to decrypt the message. We will assume that the ciphertext implicitly contains A.

Aggregate Decryption Key

Decrypt (PK, CT, and SK). The decryption algorithm takes as input the public parameters PK, a cipher text CT, which contains an access policy A, and a private key SK, which is a private key for a set S of attributes. If the set S of attributes satisfied the access structure A then the algorithm will decrypt the cipher text and return a message M.

Sharing Data

The canonical application of KAC is data sharing. The key aggregation property is especially useful when we expect the delegation to be efficient and flexible. The schemes enable a content provider to share her data in a confidential and selective way, with a fixed and small cipher text expansion, by distributing to each authorized user a single and small aggregate key.

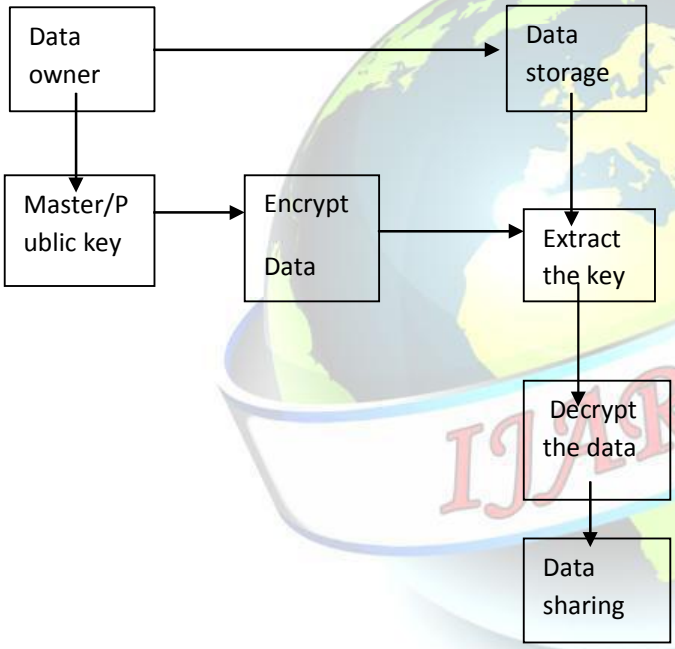
PROPOSED SYSTEM

we make a decryption key as more powerful in the sense that it allows decryption of multiple cipher texts, without increasing its size. We are introducing a public-key encryption which we call key-aggregate cryptosystem (KAC) they using AES algorithm. In KAC, users encrypt a message not only under a public-key, but also under an identifier of cipher text called class. That means the cipher texts 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 cipher text classes.

The sizes of cipher text, public-key, and 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 (but non-confidential) cloud storage. Previous results may achieve a similar property featuring a constant-size decryption key, but the classes need to conform to some pre-defined hierarchical relationship. Our work is flexible in the sense that this constraint is eliminated, that is, no special relation is required between the classes.

ARCHITECTURE



ALGORITHM EXPLANATION

In this appendix, twenty examples are provided for the MAC generation process. The underlying block cipher is either the AES algorithm or TDEA. A block cipher key is fixed for each of the currently allowed key sizes, i.e., AES-128, AES-192, AES-256, two key TDEA, and three key TDEA. For 555is given, followed by four examples of MAC generation with the key. The messages in each set of examples are derived by truncating a common fixed string of 64 bytes. All strings are represented in hexadecimal notation, with a space (or a new line) inserted every 8 symbols, for readability. As in the body of the Recommendation, K1 and K2denote the sub keys, M denotes the message, and T denotes the MAC. For the AES algorithm examples, Tlen is 128, i.e., 32 hexadecimal symbols, and K denotes the key. For the TDEA examples, Tlen is 64, i.e., 16 hexadecimal symbols, and the key, K, is the ordered triple of strings, (Key1, Key2, and Key3). For two key TDEA, Key1 = Key3. D.1 AES-128

For Examples 1–4 below, the block cipher is the AES algorithm with the following 128 bit key: K 2b7e1516 28aed2a6 abf71588 09cf4f3c.

Subkey Generation CIPHK (0 128) 7df76b0c 1ab899b3 3e42f047 b91b546f

K1 fbeed618 35713366 7c85e08f 7236a8de

K2 f7ddac30 6ae266cc f90bc11e e46d513b

The Advanced Encryption Standard (AES) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information. Encryption converts data to an unintelligible form called cipher text; decrypting the cipher text converts the data back into its original form, called plaintext. The AES algorithm is capable of using cryptographic keys of 128, 192,and 256 bits to encrypt and decrypt data in blocks of 128 bits.

RESULT

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 cipher text 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.

CONCLUSION

The implementation of the AES algorithm will provide greater security and takes less amount of time. the practical problem of privacy preserving data sharing system based on public cloud storage which requires a data owner to distribute a large number of keys to users to enable them to access his/her documents, we for the first time propose the concept of key-aggregate searchable encryption and construct a concrete KASE scheme. Both analysis and evaluation results conform that our work can provide an effective solution to building practical data sharing system based on public cloud storage

FUTURE WORK

In a KASE scheme, the owner only needs to distribute a single key to a user when sharing lots of documents with the user, and the user only needs to submit a single trapdoor when he queries over all documents shared by the same owner. However, if a user wants to query over documents shared by multiple owners, he must generate multiple trapdoors to the cloud. How to reduce the number of trapdoors under multi-owners setting is a future work. Moreover, federated clouds have attracted a lot of attention nowadays, but our KASE cannot be applied in this case directly. It is also a future work to provide the solution for KASE in the case of federated clouds.

REFERENCES

1) **Privacy-Preserving Public Auditing for Secure Cloud Storage**(Cong Wang, *Student Member, IEEE,* Sherman S.-M. Chow, Qian Wang, *Student Member, IEEE,* Kui Ren, *Member, IEEE,* and Wenjing Lou, *Member, IEEE)*

2) **Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data** (Vipul Goyal¤ Omkant Pandey*y* Amit Sahai*z* Brent Waters *x)*

3) Aggregate and Verifiably Encrypted Signatures from Bilinear Maps (Dan Bonehdabo@cs.stanford.edu,Craig Gentrycgentry@docomolabs-usa.com)