ISSN: 2278-4632
Vol-10 Issue-5 No. 14 May 2020

Developing Seclusion and Protection in Distributing Multi-authority Feature-based Encryption in Cloud Computing

D. Saritha Reddy

Assistant professor, Dept. of Master of Computer Applications, Narayana Engineering College, Gudur.

B. Jyothi

PG Scholar, Dept. of Master of Computer Applications, Narayana Engineering College, Gudur.

Abstract

Attribute-Based Encryption is used to solve the problems by sharing secret data in cloud computing. The attribute authority does not depend on central authority it can be based on decentralizing authority by using a global identifier. By using this process we can solve the problems of privacy and security. A central authority can manage only users and keys by using only the public key between each attribute authority. Once a key gives a request to outside the domain, that request can be performed by the authority in the current domain of the users. The processing of the attribute authority is based on the outside domain if the privacy and security. The decentralizing multi-authority attribute can having the both private key and public key. The author can give the request from the public key and he can get the response from the private key only. The extensibility of the authority is also supported for the privacy and security. The decentralizing multi-authority is based on the process by using dual system encryption methodology.

Key-words: In this project I can use the key words are Attribute authority, decentralizing multi-authority attribute-based encryption, dual system encryption.

2. Introduction

Cloud computing enables users to store their sensitive data into untrusted remote cloud service providers to achieve scalable services on-demand. Prominent security requirements arising from this means of data storage and management include data security and privacy of the requite use of strong encryption techniques with fine-grained access control for data security in cloud computing. Attribute-based

ISSN: 2278-4632
Vol-10 Issue-5 No. 14 May 2020

Encryption is an efficient encryption system with fine-grained access control for encrypting out-sourced data in cloud computing. With the emergence of sharing confidential corporate data on cloud servers, data are generated by several organizations, and access policies can be defined by several authorities. Single-authority ABE cannot meet the demands of decentralized distribution, and decentralizing multi-authority has been proposed to solve those problems.

For decentralizing multi-authority, the private keys of users can be generated by different authorities that do not communicate. Thus, the crucial technical challenge for decentralizing multi-authority is constructing a secret-sharing value to resist collusion attacks. The Global Identifier and central authority originated to solve the resist collusion attacks. All early schemes used central authority to deliver secret splitting, thereby assuring collision resistance under circumstances wherein authorities do not trust one another. However, a central authority should be globally trustworthy.

3. Back groundwork

Our scheme is a decentralized multi-authority that will dynamically enhance privacy and security. A central authority is not relied on to manage users and keys. Our scheme offers some improvements by combining a user's identity with the identity of the Attribute Authority where the user is located. This leads to unique user identifiers globally, and the problem of collusion resistance is also solved. In addition, user identity management does not require support from a new management organization. In our scheme, when the user requests an attribute secret key, if the attributes are located outside the domain, the request by the source in the domain to the target is used rather than by requests by users themselves. So, user identities remain private to the outside the domain, thus avoiding privacy disclosure. The key issuing protocol is simple as a result of the trust relationship of the algorithm. On the other hand, using the algorithm instead of users to initialize attribute requests can greatly improve efficiency and security. The trust relationship can also only be made by sharing the public key between each algorithm. User management and key distribution are conducted by the algorithm within the domain and therefore, the dynamic joining of the algorithm is supported in our scheme. Dual system encryption has been used to test the security of our scheme.

4. Muller Katzenbeisser Scheme

ISSN: 2278-4632
Vol-10 Issue-5 No. 14 May 2020

Muller-Katzenbeisser, and Eckert offered a different system with a centralized authority that realizes any access structure in Muller-Katzenbeisser scheme. The central authority here is mainly used to generate the public and private keys for each user and bind those keys to the identities of the users. For decryption, private keys and secret attribute keys are needed. A user's private key is generated by a central authority that is unique within the network, which ensure that the attributes are related to the same user. Thus, the full process of decryption can be performed. In addition, the collision resistance problem can be solved for each user, who applies a different relative secret attribute key from the authority.

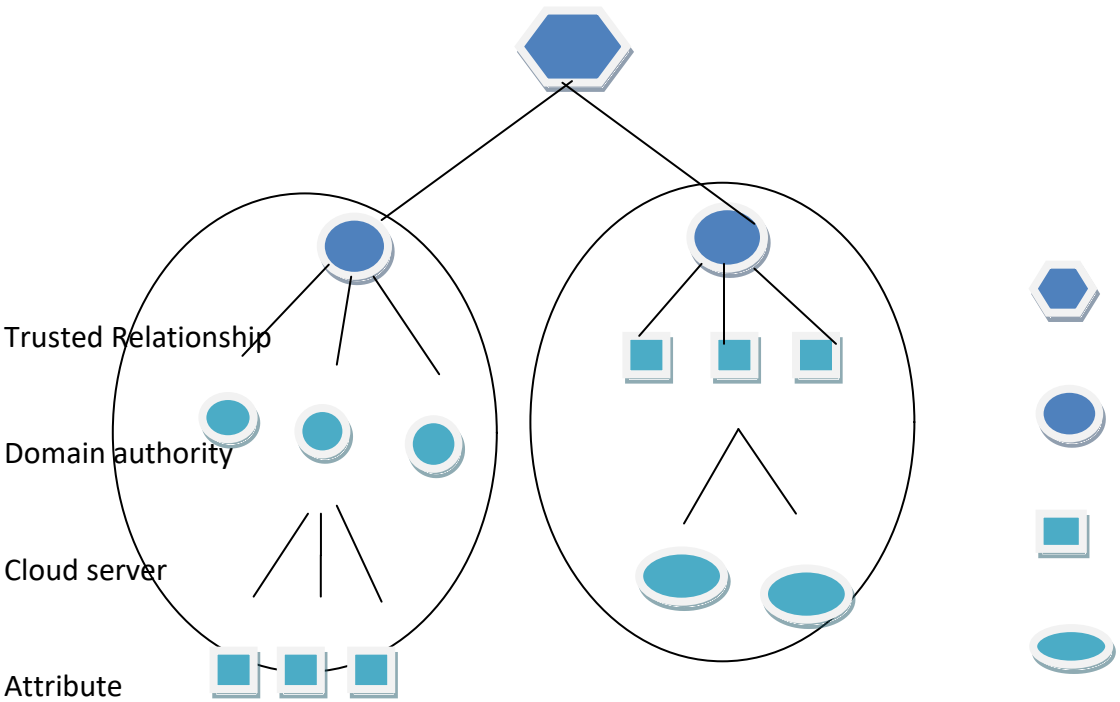
5. Lewko-Water Scheme

Lewko and Waters demonstrated a decentralizing multi-authority method that does not solely rely on the central authority of the Lewko -Waters scheme. The secret value segmentation for attributes required by access policy can be achieved using the Span Programs to construct linear secret-sharing schemes. For decryption, the required attributes should conform to the same user. The system uses to bind the different attributes together for the assigned user, and the collusion attack therefore cannot be undertaken by separate users based on their own attributes.

6. Model Definition

In cloud computing, systems are developed by information safety cooperation and university and safety association, and the data that are generated are encrypted and then stored by cloud services. The data are generated by several authorities, and the data access policy can be defined as the access policy may be changed constantly due to data issues, which may require attributes from one authority or various authorities. Assuming cooperation, universities, and safety associations are separate administer domains, and due to the collaborative work, a trust domain is constructed by those parties. In this project, an administrative domain is defined as a single authority. A trust domain is contributed by multiple administer domains, and because information is securely exchanged between the domains, cooperative work and resource sharing can be achieved.

ISSN: 2278-4632
Vol-10 Issue-5 No. 14 May 2020

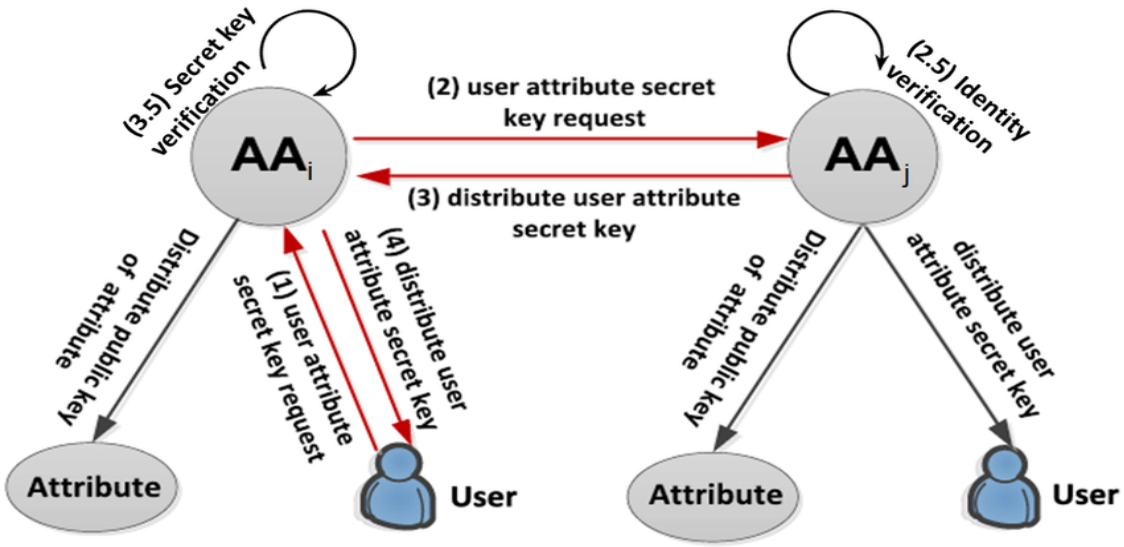


Attribute Authority Each domain administers its own users and attributes, generates the attribute public keys, and distributes the user attribute secret keys to users. Each algorithm contains its own public keys and secret keys; the public keys are used as the authentications between different algorithm's and the secret keys are used to generate the public keys of the attributes and the user attribute secret keys.

7. Flow of attribute authorization

The attributes distributed to a user might belong to different algorithm but those algorithms are based on the same trust domain. The algorithm for each administer domain can distribute user attribute secret keys for the users within and outside the domain. Because the algorithm for each administer domain knows the privilege of the users clearly, the user privileges in the administer domain are managed by the algorithm within the domain. The keys distributed to users outside the domain are based on the domain-to-domain algorithm.

ISSN: 2278-4632
Vol-10 Issue-5 No. 14 May 2020



The public keys of attributes are distributed by the algorithm within the domain. The detailed key distribution process of domain-to-domain algorithm.

8. Proposed Work

The core technique of the decentralizing multi-authority is collusion resistant; the users' keys need to be separated in multiple authorities. The secret value is sliced into private keys that are suitable for the user, and the decryption can be achieved by reconstructing the secret values of each domain and globally. This methodology of secret slicing is suitable for situations of simple access policies when attribute authorities are relatively stable. For the Lewko-Waters scheme, a secret value is sliced in the different attributes of the access policy, the access policy does not need to be considered during key distribution, and the secret share is located in the access policy of the cipher-text. Thus, Lewko-Waters scheme becomes flexible and can be changed in relation to data demands.

The privacy and security issues appear to some extent a user outside the domain requests a key from the AA directly, which will lead to issues of security and reliability for the user. In addition, working capacity will be increased dynamically. Users also need to submit their own to each authority, and therefore the authorities can obtain complete information on users according to there, which might affect their own privacies once they are used to recover the user's information.

Requests for keys outside a domain are performed by an attribute authority rather than by user requests. For that reason, the number of key applications from outside the domain will decrease sharply, and the probability of users who cheat also decreases. The public key of an attribute does not require that each attribute has a pair of random

ISSN: 2278-4632
Vol-10 Issue-5 No. 14 May 2020

numbers; only the public and secret keys of the algorithm are required, which makes the algorithm simpler, and the complexity of the system is reduced while operating.

9. Construction

The prime numbers that differ from each and represent a bilinear group of order. Two random hash functions model which we model as random oracle maps the algorithm identifier or attribute identifier to a random exponent map the algorithm identifier or user identifier to the random group elements. In addition, a finite set of hash functions needed to be defined, and hash functions are uniformly and randomly chosen from the index of a hash function is used as the secret key of the algorithm. In addition, our scheme is similar to that of, which uses composite-order bilinear groups. Here we present a multi-authority with user privacy and without the trusted authority. The requirements are non-trivial to satisfy, due in both cases to the collision resistance requirement. Brent waters suggested an approach for removing the CA requirement, in which each pair of attribute authorities would share a secret key.

Global Setup GP:

Input the security parameter generating the global parameter GP. A bilinear group of order is chosen. GP are and a generator. The random oracle functions are also included.

Authority Setup:

Input GP and the algorithm identity, and each algorithm chosen a random exponent. In addition, a hash function is uniformly and randomly chosen from a finite set of hash functions. The index of the hash function in the function set keeps the secret key of the algorithm. Therefore, the secret key of an algorithm can be represented as, and the public key is expressed as where the identity of the attribute authority.

Request Attribute:

The public key of an attribute is generated by an algorithm according to the attribute identifier, and the secret key of represents algorithm. Where, Attribute identifier consists of the algorithm identity and the attribute identity inside the domain. The attribute identity represents inside the current domain. The attribute identifier is unique within the entire trust domain therefore, the public key of each attribute can also be considered to be unique within the entire trust domain. In addition, the public key of the attribute is generated by the algorithm's secret key, which will also ensure the reliability of the public key.

Key Gen User Attribute:

The algorithm is used to generate the user attribute secret key by an algorithm outside the domain. If the requested attributes by the user is outside the domain, the algorithm

ISSN: 2278-4632
Vol-10 Issue-5 No. 14 May 2020

of the current domain will initiate the request to the target algorithm, the request information includes instead of for issues of privacy. The target domain will check the identity of the current algorithm, and once it is shown to be trusted, the user attribute secret keys will be generated. When the current algorithm obtains the keys, it will check the validity of the keys using the public keys of the requested attributes. The detailed process is described in the Key issuing protocol.

Encrypt:

The encrypted algorithm inputs message, access matrix .maps the row of matrix to the attributes, and the relative public keys are requested by the access matrix. A random is picked, and a random vector is picked that has as its first entry.

Decrypt:

The decryption algorithm inputs cipher-text and the user attribute secret key set for one user. Decryption will occur once the user has the requested user attribute secret keys that satisfied the access matrix during the encryption. Assume the cipher-text is encrypted under the access matrix. To decrypt the message and can be computed according to the random function, where is the algorithm identity at the user's location.

10. Comparison of Different Schemes

Scheme	AA Key Size	User Key Size	Ciphertext	GID	Key Issuing protocol	Trust Relationship
Chase	AK+N	Au+1	Ac+2	privacy	(P+3) Au	N(N-1)/2
LewkoWaters	2 Ak	Au	3 Ac+1	public		N
Rahulamathavan-Veluru	Ak+2	Au+2N	Ac+3	privacy	(P+2) Au	N
Ours	N+1	Au	4Ac+1	privacy	2Au	N

User identity management all need to be offered by related organizations in all the above schemes without central authority. In our scheme, each algorithm has public and secret keys, a private key of an attribute do not require, which reduce the quantity of key. Only when the system is set up, the public key of each algorithm and the basic parameters are distributed, which simplifies the process of trust establishment. The key issuing protocol for privacy, only need to use the public key of an algorithm to realize the trust algorithm, does not require support from the 2PC protocol, no new parameters need to be generated.

ISSN: 2278-4632
Vol-10 Issue-5 No. 14 May 2020

11. Conclusion

Decentralizing multi-authority can solve problems arising from security requirements of sharing confidential corporate data on cloud servers. For decentralized multi-authority ABE schemes with non-central authority, the collusion resistance can be solved using the GID. Therefore, the uniqueness of user identities needs to be managed globally, which results in crucial problems of privacy and security. In this essay, a scheme without a central authority to manage keys and users has been proposed, and privacy and security have been enhanced dynamically.

References

1. J. Horwitz, B. Lynn, “Towards hierarchical identity-based encryption,” in Proc. EUROCRYPT, Amsterdam, The Netherlands, April. 2002, pp. 466-481.
2. C. Gentry, A. Silverberg, “Hierarchical ID-based cryptography,” in Proc. ASIACRYPT, Singapore, December. 2002, pp. 548-566.
3. D. Boneh, X. Boyen, "Efficient Selective-ID secure identity-based encryption without random oracles," in Proc. EUROCRYPT, Interlaken, Switzerland, May. 2004, pp. 223-238.
4. Rajasekar, P. and Mangalam, D. (2016) Efficient FPGA implementation of AES 128 bit for IEEE 802.16e mobile WiMax standards. Circuits and Systems, 7, 371-380. doi: 10.4236/cs.2016.74032.
5. D. Boneh, X. Boyen, E.Goh, "Hierarchical identity-based encryption with constant size ciphertext,” in Proc. EUROCRYPT, Aarhus, Denmark, May. 2005, pp. 440-456.
6. Penchalaiah P, Ramesh Reddy K, “Secure and Cost Effective Cryptosystem Design Based on Random Multiple Key Streams”, Journal of Information Security Research,(ISSN: 0976-4143) DIRF Publisher, Volume 7, Number 1, pp. 29-40, March 2016.
7. X. Boyen, B. Waters, “Anonymous hierarchical identity-based encryption (without random oracles),” in Proc. CRYPTO, Santa Barbara, California, USA, August. 2006, pp. 290-307.

Author’s Profile

D. Saritha has received her degree from Sri Venkateswara University (SVU) in 2008 and completed M-Tech from Narayana

ISSN: 2278-4632
Vol-10 Issue-5 No. 14 May 2020

University in 2010. She is defined to the teaching field for the last 9 years and working as an assistant professor at Narayana Engineering college-Gudur and also now persuing Ph.D. Area with Computer Networks.

B. Jyothi has received her B.Sc. Computers from Vidyalaya Degree College Gudur, affiliated to VSU Nellore in 2018 and pursuing M.C.A at Narayana Engineering College (NECG), GUDUR, AP affiliated to JNTUA in (2018-2020).

ISSN: 2278-4632
Vol-10 Issue-5 No. 14 May 2020

