PalArch's Journal of Archaeology of Egypt / Egyptology

DATA SECURITY MODEL IN CLOUD USING HECC FOR KEYWORD BASED SEARCH

Devi. T*[1], Ganesan. R[2]

***[1]Assistant Professor, Saveetha School of Engineering, SIMATS, Chennai.**

[2]Professor, SCSE, Vellore Institute of Technology, Chennai.

Devi. T , Ganesan. R , Data Security Model In Cloud Using Hecc For Keyword Based Search , Palarch's Journal Of Archaeology Of Egypt/Egyptology 18(7), 3374-3381. ISSN 1567-214x.

Keywords : AWS, HECC, Cloud Computing, Encryption/Decryption.

Abstract:

Cloud computing is a easiest way to share a data in various users. Nowadays cloud storage become more popular but the user can share any sensitive data to a cloud server should be treated as untrusted entity. Our proposed work is each and every data indexed in keyword should be generated after encrypted file. A key values are stored in index some authorized user wants to identify the data using keyword in cloud to improve the performance time in HECC algorithm. Hence, it is necessary to implement Data Encryption and Decryption scheme with cloud storage to provide automatic encryption/decryption in AWS cloud service. The purpose of using HECC for encryption/decryption is that it is secure and high computational complexity.

1. Introduction:

Cloud computing could also be a general term for love or money that involves delivering hosted services over the online. These services are separated into three main branches: infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS). A cloud is often private or public. A public cloud sells services to anyone on the online. a private cloud could also be a proprietary network or a knowledge centre that supplies hosted services to a limited number of people, with certain access and permissions settings. Private or public, the goal of cloud computing is to provide easy, scalable access to computing resources and IT services. Cloud infrastructure involves the hardware and software components required for proper implementation of a cloud computing model. Cloud computing can also be thought of as utility computing, or on-demand computing. Cloud computing works by allowing client devices to access data over the online, from remote servers, databases

and computers. An internet network connection links the front (includes the accessing client device, browser, network and cloud software applications) with the rear end, which consists of databases, servers and computers). The back-end functions as a repository, storing data that's accessed by the front. Communications between the front and back ends are managed by a central server. The central server relies on protocols to facilitate the exchange of data. The central server uses both software and middleware to manage connectivity between different client devices and cloud servers. Typically, there will be an obsessive server for each individual application.

Infrastructure as a service (IaaS)

IaaS is that the commonest initiative on the journey to the cloud for a corporation with an existing IT estate. IaaS essentially allows companies to rent the remote physical and logical (networking) IT infrastructure from a cloud vendor and deploy virtual machines (VMs) which they will then manage. The management of public cloud VMs is analogous to once they are hosted on premise, however the necessity to have or manage any of the physical environment is totally removed. due to this, existing IT skills are mostly transferable and therefore the pay-as-you-go nature of the general public cloud can make this a comparatively painless and price effective thanks to make the primary steps in to the cloud.

Platform as a service (PaaS)

PaaS enables consumers to make environments on-demand for developing, testing, delivering and managing software and services. this is often a particularly efficient way of making new web or mobile apps, databases and container clusters as administrators don't got to worry about creating or managing the broader infrastructure

Software as a service (SaaS)

SaaS may be a simplistic way of granting end users access to ready-to-use software. instead of having to deploy, patch and update variety of apps on a spread of devices. SaaS applications are cloud based then are often reached over the web using just a contemporary browser. This method of accessing and using software means potentially all required user apps are often accessed from an equivalent user dashboard through a browser

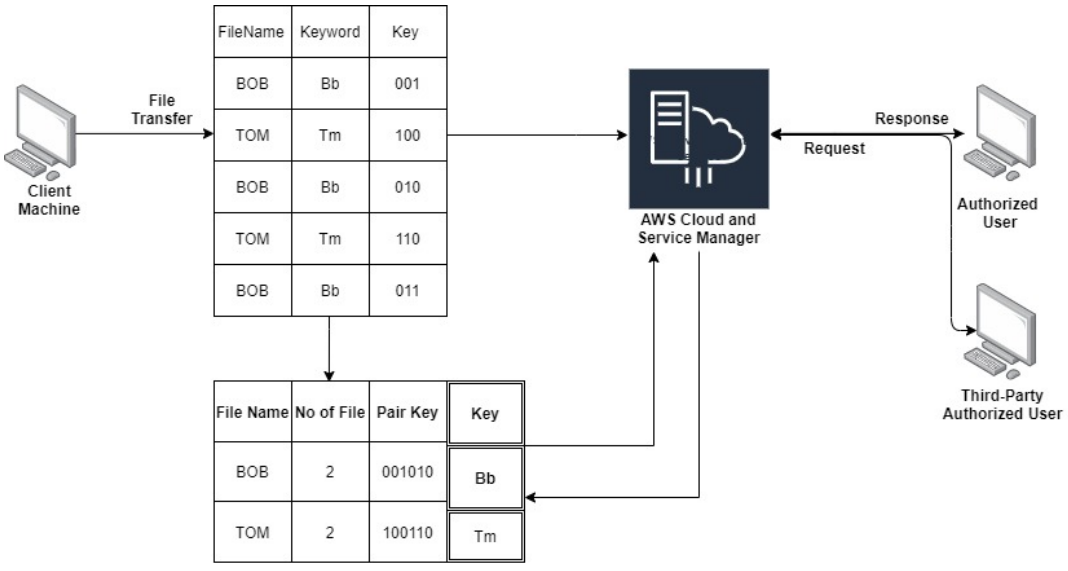

Fig 1.0 Shows client machine share the data after encrypted file index should be created with keyword and key value are stored in AWS cloud storage. Authorized user can pair the key to decrypt the file.

2. Data security model:

Data Security in Cloud Computing Data protection may be a crucial security issue for many organizations. Before getting into the cloud, cloud users got to clearly identify data objects to be protected and classify data supported their implication on security, then define the safety policy for data protection also because the policy enforcement mechanisms. for many applications, data objects would come with not only bulky data at rest in cloud servers (e.g., user database and/or filesystem), but also data in transit between the cloud and therefore the user(s) which might be transmitted over the web or via mobile media (In many circumstances, it might be less expensive and convenient to maneuver large volumes of knowledge to the cloud by mobile media like archive tapes than transmitting over the web.). Data objects can also include user identity information created by the user management model, service audit data produced by the auditing model, service profile information want to describe the service instance(s), temporary runtime data generated by the instance(s), and lots of other application data. differing types of knowledge would be of various value and hence have different security implication to cloud users. for instance, user database at rest in cloud servers could also be of the core value for cloud users and thus require strong protection to ensure data confidentiality, integrity and availability. User identity information can contain Personally Identifiable Information (PII) and has impact on user privacy. Therefore, just authorized users should be allowed to access user identity information. Service audit data provide the evidences associated with compliances and therefore the fulfillment of Service Level Agreement (SLA), and will not be maliciously manipulated. Service profile information could help attackers locate and identify the service instances and will be protected. Temporary runtime data may contain critical data associated with user business and will be segregated during runtime and securely destroyed after runtime.

Data confidentiality assurance:

This service protects data from being disclosed to illegitimate parties. In Cloud Computing, data confidentiality may be a basic Security Service to be in place. Although different applications may have different requirements in terms of what quite data need confidentiality protection.

Data integrity protection:

This service protects data from malicious modification. When having outsource their data to remote cloud servers, cloud users must have the way to see whether or not their data at rest or in transit are intact. Such a Security Service would be of the core value to cloud users. When auditing cloud services, it is also critical to make sure that everyone the audit data are authentic since these data would be of legal concerns.

Guarantee of data availability:

This service assures that data stored within the cloud are available on each user retrieval request. This service is especially important for data at rest in cloud servers and associated with the fulfillment of Service Level Agreement. For long-term data storage services, data availability assurance is of more importance due to the increasing possibility of knowledge damage or loss over the time.

Secure data access:
This Security Service is to limit the disclosure of knowledge content to authorized users. In practical applications, disclosing application data to unauthorized users may threat the cloud user's business goal. In missioncritical applications, inappropriate disclosure of sensitive data can have juristic concerns. For better protection on sensitive data, cloud users may have finegrained data access control within the sense that different users may have access to different set of knowledge .

Regulations and compliance:
In application scenarios, storage and access of sensitive data may need to comply specific compliance. for instance, disclosure of health records could also be limited by the insurance Portability and Accountability Act (HIPAA) additionally to the present, the geographic location of knowledge would frequently be of concern thanks to export-law violation issues. Cloud users should thoroughly review these regulation and compliance issues before moving their data into the cloud.

Service audition:
This service provides how for cloud users to observe how their data are accessed and is critical for compliance enforcement. within the case of local storage, it isn't hard to audit the system. In Cloud Computing, however, it requires the service provider to support trustworthy transparency of data access.

3. Key Agreement:

The client implements the HECC algorithm to get the general public and personal keys. The key pair is defined as {pk, d} where pk - public key and d - private key. The Diffie-Hellman key agreement technique was designed for the multiplicative group of numbers, but it can adjusted easily to general groups. allow us to consider G be a gaggle whose elements are often efficiently represented the group operation scan be efficiently evaluated also . The group is jacobians of hyper elliptic curves.

Algorithm:

1. **Upload Encrypted file in cloud. Each and every file should be upload by the client and generating the encryption keys**.

Step 1: File upload by client
Step 2: Each new file generate public key in Index
P_k (K_n,K_d) Files(F_d)
Set of files like
F_d=(f_{d1}, f_{d2}, f_{d3}…..f_{dn}}

Step 3: Encrypt the file using keyword and public key

$ENC(W_n \| K_n)$

Kn : Key

$ENC\ K_n(FD)$ // File Encryption

2. After encrypted file client side keyword should be indexed on file:

Step 1: Once file encryption is done after client will set keyword

Step 2: Authorized user search the keyword

$S = (S_1,S_2,S_n)=ENC\ K_n(W_n||1),ENC\ K_n(W_n||2),\ldots ENC\ K_n(W_n||n)$

So S is Search, W_n is a keyword.

3. Cloud service manager check the authorized user and decrypt the file:

Step 1 : Service manager send the decryption file to autoized user when paring key matched.

$DEC\ K_d(F_d)$

F_d is File decrypted using second key (K_d)

Encryption and Decryption:

The data owner will encrypt the files to the cloud before sending with the general public key pk - -> E. The hash value generated is stored for further verification process and therefore the data encrypted is uploaded to the cloud. If A wants to send data message M to B, it does the subsequent

- It obtains the public key pk of receiver B.
- It chooses a secret number a [1,r-1]
- Computes the value C1 = aR.
- Compute the value C2 = M + a(pk).
- Send (C1, C2) to B.

the info user must access the file, a download request are going to be sent to the cloud and a decryption key's wont to decrypt the retrieved content. After retrieval of content, the hash value are going to be calculated again. The file integrity can now be verified upon comparison. because the file is stored within the cloud, the comparison of hash value are going to be helpful to spot whether the file is ideal while it's stored within the cloud. The “receiver B can decrypt the cloud data by doing the following

- Receive the encrypted message (C1, C2) from sender A
- Compute the message value M = C2 − bC1”.

4. Comparison and Analysis:

A. PERFORMANCE ANALYSIS OF ECC AND HECC

File Size	ECC Algorithm (160 bits) (Hussain et al.,)	HECC Algoithm(80 bits)
10Kb	621	358
20Kb	544	428
30Kb	681	514
40Kb	774	674

B. PERFORMANCE ANALYSIS of Search Keyword time

	ECC Algorithm (160 bits) (Hussain et al.,)		HECC Algoithm(80 bits)	
File Size	Encryption Time (ms)	Decryption Time(ms)	Encryption Time (ms)	Decryption Time(ms)
10Kb	1558	1234	942	685
20Kb	1368	1121	881	611
30Kb	1314	1011	753	512
40Kb	1237	988	684	422

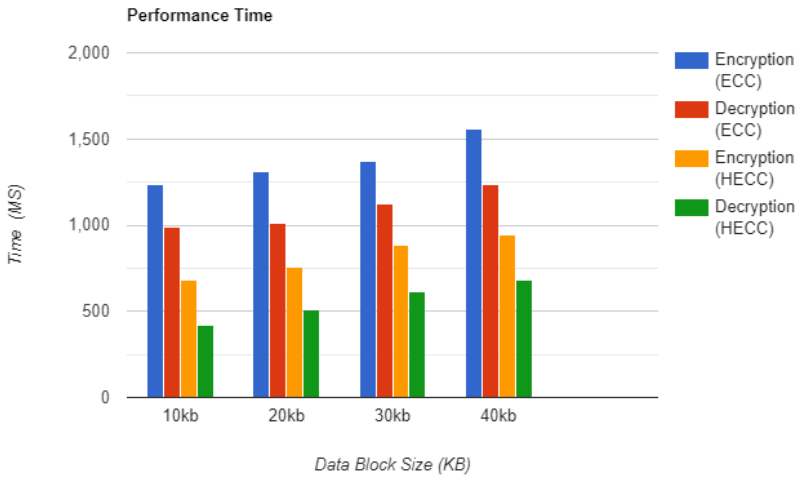

Fig 1.1 shows the time comparison of Data block size and Time using HECC algorithm

5. Conclusion and Future Work:

New threats are being discovered every day so making an application to keep your data secure from such threats is important. The proposed approach aims to provide user satisfaction of storage personal and sensitive data including files and images securely. This paper shows successfully implementation of encryption/decryption to automatic encryption of hidden confidential in cloud.

The paper shows the successfully implementation of HECC algorithm for automatic encryption and decryption for confidential file present in AWS cloud. The result obtained shows to improve the performance time of encryption and decryption in HECC Algorithm in future to keyword search scheme with assigned proper roles to delegate data based search capabilities over encrypted data to cloud provider in efficient time.

Reference:

1. J. Wei, W. Liu and X. Hu, "Secure data sharing in cloud computing using revocable-storage identity-based encryption", IEEE Trans. Cloud Computing, vol. 6, no. 4, pp. 1136-1148, 2018.
2. A. Boldyreva, V. Goyal and V. Kumar, "Identity-based encryption with efficient revocation", Proc. ACM Conf. Comput. Commun. Secur., pp. 417-426, 2008.
3. A. Sahai, H. Seyalioglu and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption", Proc. Annu. Cryptol. Conf., pp. 199-217, 2012.
4. K. Lee, S. G. Choi, D. H. Lee, J. H. Park and M. Yung, "Self-updatable encryption: Time constrained access control with hidden attributes and better efficiency", Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur., pp. 235-254, 2013.
5. K. Lee, "Self-updatable encryption with short public parameters and its extensions", Des. Codes Cryptogr., vol. 79, no. 1, pp. 121-161, 2016.
6. L. M. Vaquero, L. Rodero-Merino, J. Caceres and M. Lindner, "A break in the clouds: Towards a cloud definition", ACM SIGCOMM Comput. Commun. Rev., vol. 39, no. 1, pp. 50-55, 2008.
7. K. Chard, K. Bubendorfer, S. Caton and O. F. Rana, "Social cloud computing: A vision for socially motivated resource sharing", IEEE Trans. Serv. Comput., vol. 5, no. 4, pp. 551-563, Oct.-Dec. 2012.
8. C. Wang, S. S. Chow, Q. Wang, K. Ren and W. Lou, "Privacy-preserving public auditing for secure cloud storage", IEEE Trans. Comput., vol. 62, no. 2, pp. 362-375, Feb. 2013.
9. G. Anthes, "Security in the cloud", Commun. ACM, vol. 53, no. 11, pp. 16-18, 2010.
10. K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing", IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 9, pp. 1717-1726, Sep. 2013.
11. B. Wang, B. Li and H. Li, "Public auditing for shared data with efficient user revocation in the cloud", IEEE Trans. Serv. Comput., vol. 8, no. 1, pp. 92-106, Jan./Feb. 2015.
12. S. Ruj, M. Stojmenovic and A. Nayak, "Decentralized access control with anonymous authentication of data stored in clouds", IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 2, pp. 384-394, Feb. 2014.
13. X. Huang, J. Liu, S. Tang, Y. Xiang, K. Liang, L. Xu, et al., "Cost-effective authentic and anonymous data sharing with forward security", IEEE Trans. Comput., vol. 64, no. 4, pp. 971-983, Apr. 2015.
14. C.-K. Chu, S. S. Chow, W.-G. Tzeng, J. Zhou and R. H. Deng, "Key-aggregate cryptosystem for scalable data sharing in cloud storage", IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 2, pp. 468-477, Feb. 2014.
15. D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing", SIAM J. Comput., vol. 32, no. 3, pp. 586-615, 2003.

16. V. Goyal, "Certificate revocation using fine grained certificate space partitioning", Proc. 11th Int. Conf. Financial Cryptography, pp. 247-259, 2007.
17. B. Libert and D. Vergnaud, "Adaptive-id secure revocable identity-based encryption", Proc. Cryptographers Track RSA Conf., pp. 1-15, 2009.
18. B. Libert and D. Vergnaud, "Towards black-box accountable authority IBE with short ciphertexts and private keys", Proc. 12th Int. Conf. Practice Theory Public Key Cryptography, pp. 235-255, 2009.
19. J. Chen, H. W. Lim, S. Ling, H. Wang and K. Nguyen, "Revocable identity-based encryption from lattices", Proc. 17th Australasian Conf. Inf. Security Privacy, pp. 390-403, 2012.
20. J. H. Seo and K. Emura, "Revocable identity-based encryption revisited: Security model and construction", Proc. 16th Int. Conf. Practice Theory Public-Key Cryptography, pp. 216-234, 2013.