SecureCloud Access: Enhanced Data Control and Expressive Data Management using crypto+

Sri Varsha Shwetha M[1] ,Dhanushya M[2], Preethi N[3], Roshni R[4],Soniya J [5]

[1] *Assistant Professor,Dept of Information of Technology,Meenakshi College Of Engineering*
[2,3,4,5] *Students of ,Dept of Information of Technology,Meenakshi College Of Engineering*

---***---

Abstract - Secure cloud storage, which is an emerging cloud service, is designed to protect the confidentiality of outsourced data but also to provide flexible data access for cloud users whose data is out of physical control. Cipher text Policy Attribute-Based Encryption (CP-ABE) is regarded as one of the most promising techniques that may be leveraged to secure the guarantee of the service. However, the use of CP-ABE may yield an inevitable security breach which is known as the misuse of access credential (i.e. decryption rights),due to the intrinsic "all-or-nothing" decryption feature of CP-ABE. In this paper, we investigate the two main cases of access credential misuse: one is on the semi-trusted authority side, and the other is on the side of cloud user. To mitigate the misuse, we propose the first accountable authority and revocable CP-ABE based cloud storage system with white-box traceability and auditing, referred to as Crypt Cloud. We also present the security analysis and further demonstrate the utility of our system via experiments.

***Key Words*: Secure cloud storage, CP-ABE (Ciphertext-Policy Attribute-Based Encryption), authority, Whitebox, Auditing, CryptCloud+**

1.INTRODUCTION

Data owners will store their data in public cloud along with encryption and particular set of attributes to access control on the cloud data. While uploading the data into public cloud they will assign some attribute set to their data. If any authorized cloud user wants to download their data they should enter that particular attribute set to perform further actions on data owner's data. A cloud user wants to register their details under cloud organization to access the data owner's data. Users want to submit their details as attributes along with their designation. Based on the user details Semi-Trusted Authority generates decryption keys to get control on owner's data. An user can perform a lot of operations over the cloud data. If the user wants to read the cloud data he needs to be entering some read related attributes, and if he wants to write the data he needs to be entering write related attributes. For each and every action user in an organization would be verified with their unique attribute set. These attributes would be shared by the admins to the authorized users in cloud organization. These attributes will be stored in the policy files in a cloud. If any user leaks their unique decryption key to the any malicious user data owners wants to trace by sending audit request to auditor and auditor will process the data owners request and concludes that who is the guilty.

1. Organization profile creation & Key Generation

User has an initial level Registration Process at the web end. The users provide their own personal information for this process. The server in turn stores the information in its database. Now the Accountable STA (semi-trusted Authority) generates decryption keys to the users based on their Attributes Set (e.g. name, mail-id, contact number etc...). User gets the provenance to access the Organization data after getting decryption keys from Accountable STA

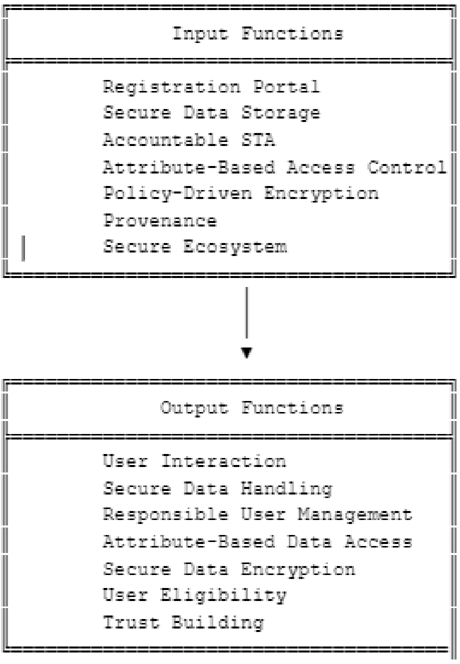


Fig.1 Organization profile creation & Key Generation

2. Data Owners File Upload

The data owners create their accounts under the public cloud and upload their data into public cloud. While uploading the files into public cloud data owners will encrypt their data using RSA Encryption algorithm and generates public key and secret key. And also generates one unique file access permission key for the users under the organization to access their data.

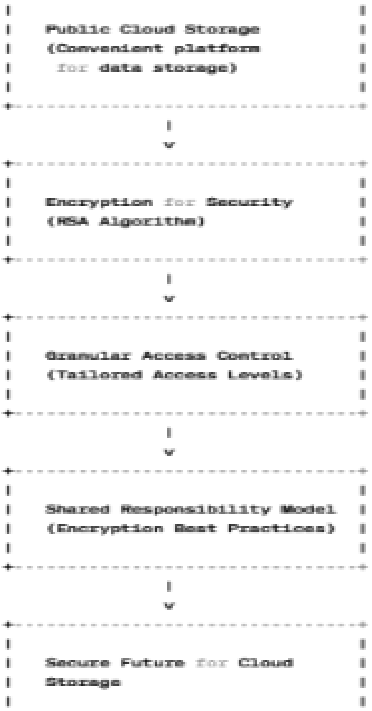


Fig.2 Data Owners File Upload

3. File Permission & Policy File Creation

Different data owners will generate different file permission keys to their files and issues those keys to users under the organization to access their files. And also generates policy files to their data that who can access their data. Policy File will split the key for read the file, write the file, download the file and delete the file.

3.1 Policy Files and Attribute-Based Encryption (ABE):

To further refine access control, data owners create policy files associated with their data. These policy files function like access control lists (ACLs) but leverage a more sophisticated approach called Attribute-Based Encryption (ABE). ABE enables specifying access permissions based on user attributes rather than just identities.

3.2 KEY SPLITTING VS. ATTRIBUTE-BASED ENCRYPTION:

Splitting the key is a simplified explanation. In reality, ABE employs more intricate cryptographic techniques. The policy file likely defines how user attributes correspond to specific decryption keys or key components. Users only receive the key components necessary for their authorized actions based on their attributes. This is the technical aspects behind the file access control system. It introduces Attribute-Based Encryption and explains how user attributes determine access permissions.

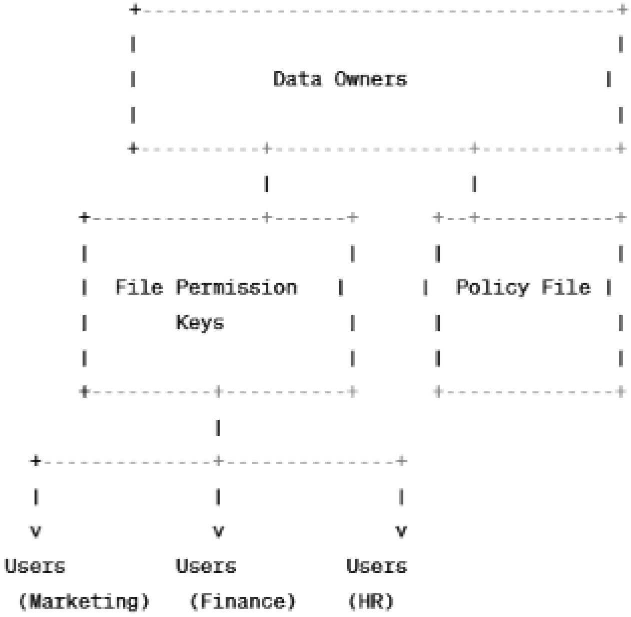


Fig.3 File Permission & Policy File Creation

4. Tracing who is guilty

Authorized DUs are able to access (e.g. read, write, download, delete and decrypt) the outsourced data. Here file permission keys are issued to the employees in the organization based on their experience and position. Senior Employees have all the permission to access the files (read, write, delete, & download). Fresher's only having the permission to read the files. Some Employees have the permission to read and write. And some employees have all the permissions except delete the data. If any Senior Employee leaks or shares their secret permission keys to their junior employees they will request to download or delete the Data Owners Data. While entering the key system will generate attribute set for their role in background validate that the user has all rights to access the data. If the attributes set is not matched to the Data Owners policy files they will be claimed as guilty. If we ask them we will find who leaked the key to the junior employees. If a Senior Employee shares their keys, junior employees may attempt unauthorized actions.Senior Employees have all permissions, while Freshers only have read permissions.File permission keys are issued based on employee experience and position.

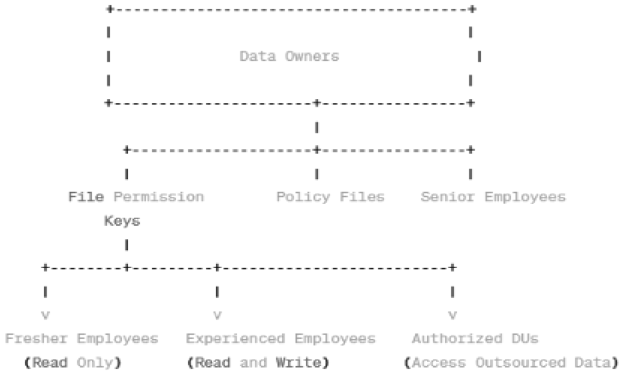


Fig.4 Tracing who is guilty

6.Algorithm

6.1 RSA Algorithm

RSA (Rivest-Shamir-Adleman) algorithm is a widely used asymmetric cryptographic algorithm for secure data transmission.

A. Key Generation:

- Generate a pair of RSA keys: a public key and a private key.
- The public key will be used for encryption, and it can be freely distributed.
- The private key must be kept secret and securely stored, as it is used for decryption.

I. Choose two distinct prime numbers, pp and qq.

II. Compute $n=p\times qn=p\times q$.

III. Compute $\phi(n)=(p-1)\times(q-1)\phi(n)=(p-1)\times(q-1)$, where $\phi\phi$ is Euler's totient function.

IV. Choose an integer ee such that $1<e<\phi(n)1<e<\phi(n)$ and ee is coprime to $\phi(n)\phi(n)$ (i.e., $\gcd(e,\phi(n))=1\gcd(e,\phi(n))=1$).

V. Compute dd as the modular multiplicative inverse of ee modulo $(n)\phi(n)$, i.e., $d\times e\equiv 1 \bmod \phi(n) d\times e\equiv 1 \bmod \phi(n)$.

VI. The public key is $(n,)(n,e)$ and the private key is $(n,d)(n,d)$.

B. Encryption:

- When a user wants to send data to the cloud, they use the recipient's public key to encrypt the data.
- The encrypted data is then sent to the cloud for storage or processing.

I. Convert the plaintext message MM into an integer mm such that $0\leq m<n0\leq m<n$.

II. Compute the ciphertext cc using the public key $(n,)(n,e)$ and the formula $c\equiv me\text{mod } nc\equiv me\text{mod}n$.

C. Decryption:

- When the cloud needs to send encrypted data back to a user, it uses the recipient's private key to decrypt the data.
- Only the user with the corresponding private key can decrypt the data successfully.
- Given the ciphertext cc, compute the plaintext mm using the private key $(n,)(n,d)$ and the formula $m\equiv cd\text{mod } nm\equiv cd\text{mod}n$.

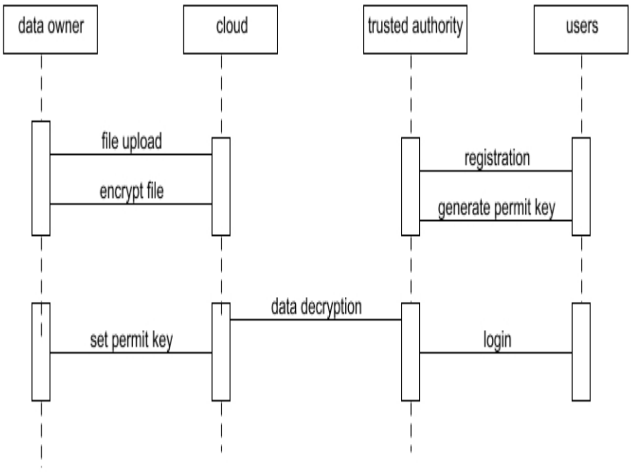


Fig.6.1 Flow Diagram

6.2 Crypto+ Algorithm

Cryptographic algorithm for securing cloud data in an office environment requires careful consideration of various factors such as encryption strength, key management, and data integrity. Let's design a simple algorithm called "CloudSecure+" (Crypto+) using a combination of symmetric and asymmetric encryption along with hashing for data integrity.

A. Key Generation:

- Generate a symmetric encryption key (SK) for encrypting data locally on the office machines.

- Generate an asymmetric key pair (public key, PK and private key, PrK) for secure communication between the office machines and the cloud server.

B. Data Encryption (Local):

- Encrypt data locally using symmetric encryption (e.g., AES) with the symmetric key (SK).

EncryptedData = AES_Encrypt(Data, SK)

C. Secure Transmission:

- Encrypt the symmetric key (SK) with the public key (PK) for secure transmission to the cloud server. **EncryptedSK = RSA_Encrypt(SK, PK)**

D. Data Transmission:

- Transmit the encrypted data and the encrypted symmetric key to the cloud server.

E. Data Storage (Cloud):

- Store the encrypted data and the encrypted symmetric key securely in the cloud.

- Optionally, store a cryptographic hash of the encrypted data for data integrity verification.

F. Data Retrieval (Cloud):

- Retrieve the encrypted data and the encrypted symmetric key from the cloud server.

G. Key Decryption (Local):

- Decrypt the symmetric key (SK) using the private key (PrK).

SK = RSA_Decrypt(EncryptedSK, PrK)

H. Data Decryption (Local):

- Decrypt the data locally using the decrypted symmetric key (SK).

DecryptedData = AES_Decrypt(EncryptedData, SK)

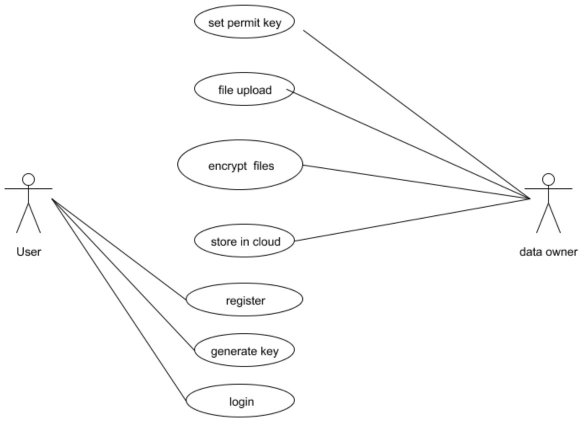


Fig. **6.2 Crypto+ Algorithm**

7. Implementation

First,the ower want to create the account or login to upload,delete,for knowing employee detail

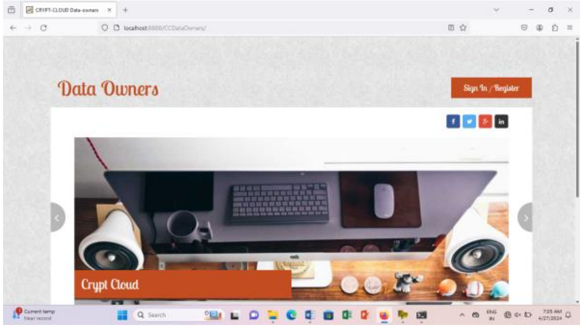


Fig.7.1 Data Owner

It is the page after the data ower get login.Mainly data ower used for finding employee who misuse the file and to protect from leaking the information.

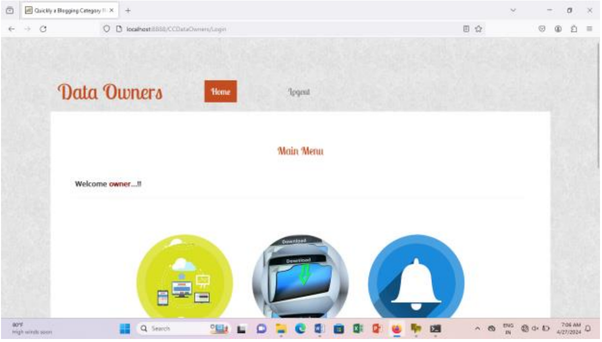
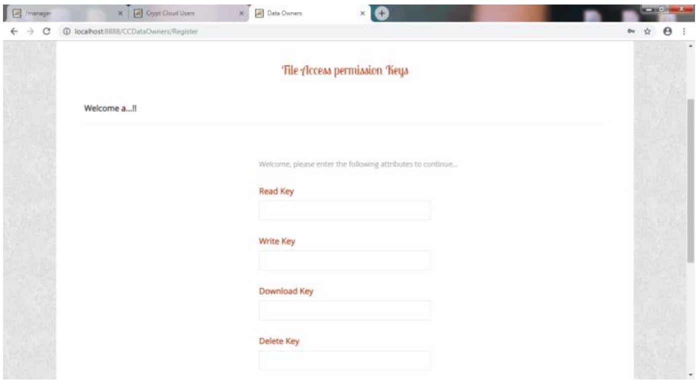


Fig.7.2 Data ower home page

The file are protected by the way of generate separate key for each access like read, write, download, delete.



File Access permission Keys

Welcome a...!!

Welcome, please enter the following attributes to continue...

Read Key

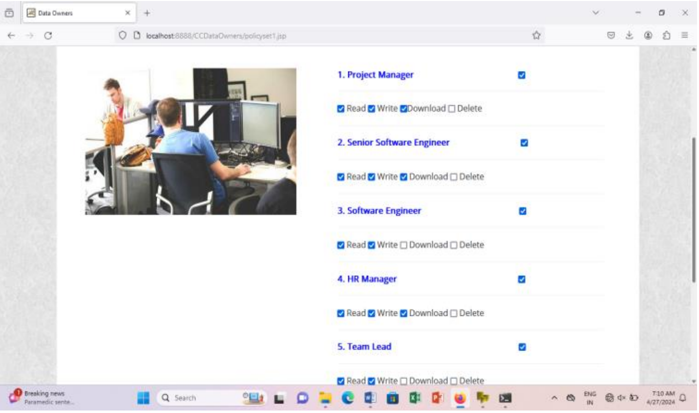
Write Key

Download Key

Delete Key

Fig.7.3 File Access Permission Key

The permission for the employee who access the file (read, write, download, delete).



1. Project Manager ☑

☑ Read ☑ Write ☑ Download □ Delete

2. Senior Software Engineer ☑

☑ Read ☑ Write ☑ Download □ Delete

3. Software Engineer ☑

☑ Read ☑ Write □ Download □ Delete

4. HR Manager ☑

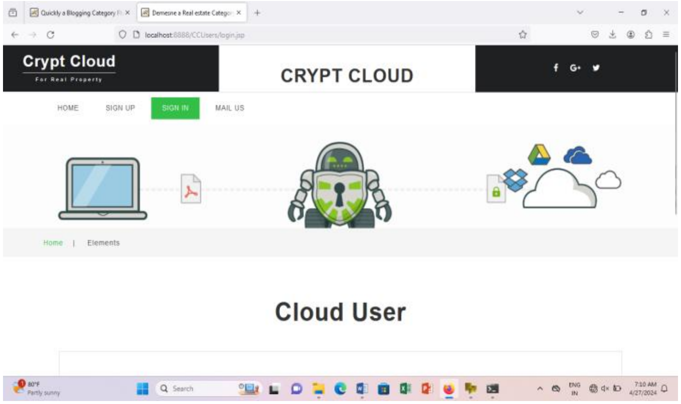
☑ Read ☑ Write ☑ Download □ Delete

5. Team Lead ☑

☑ Read ☑ Write □ Download □ Delete

Fig.7.4 File Policy

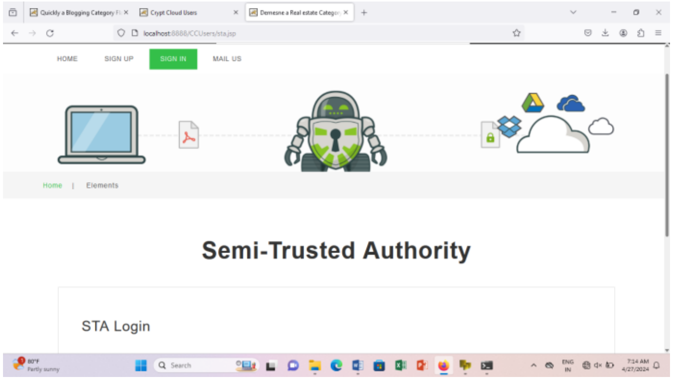
Now the user(employee)can login with there Cloud user account either by creating the account nor by register the account



Cloud User

Fig.7.5 Cloud User

STA(Semi Trusted Authority) its used to generate the attribute for cloud user(employee)

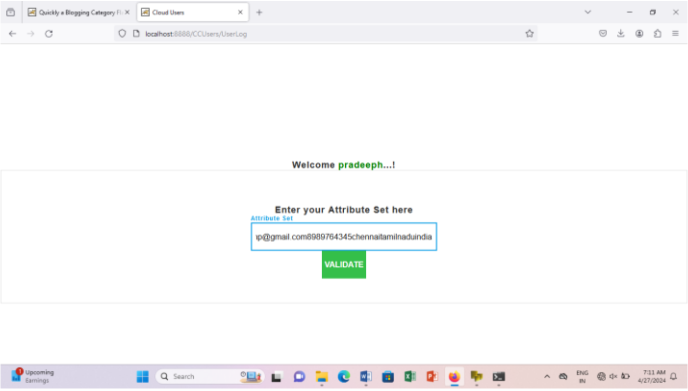


Semi-Trusted Authority

STA Login

Fig.7.6 Cloud User

Now we can enter the attributes value in attribute set field(cloud user page)by the cloud user



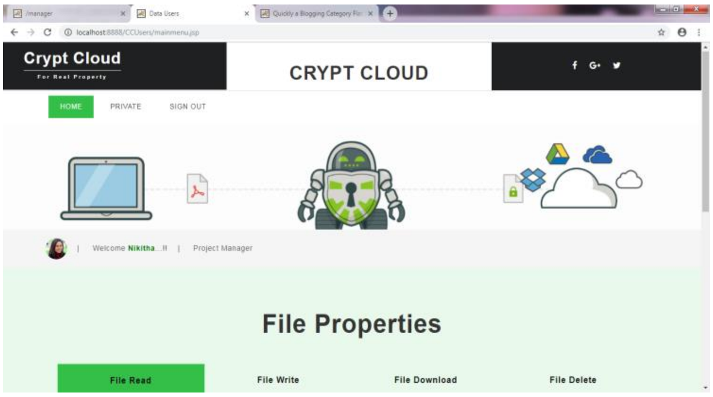
Welcome pradeeph...!

Enter your Attribute Set here

Attribute Set: ıp@gmail.com8989764345chennaitamilnaduindia

Fig.7.7 Attribute Set

Now cloud user can access the file with there given permission.



Welcome Nikitha...!! | Project Manager

File Properties

Fig.7.7 File Properties

8.Future Enchancement

CryptoCloud+ to provide partial and fully public traceability represents a significant step towards achieving secure and transparent cloud computing environments. By addressing challenges related to performance, privacy, scalability, and data integrity, this extension enhances the overall effectiveness of CryptoCloud+ in meeting the evolving needs of modern cloud users. Through efficient implementation of traceability features and careful consideration of their implications, CryptoCloud+ solidifies its position as a leading solution for secure and transparent cloud computing. As technology continues to evolve, the need for secure and transparent cloud solutions becomes increasingly apparent. In line with this, our future work aims to extend CryptoCloud+ to offer partial and fully public traceability features while maintaining high performance. This expansion addresses the growing demand for accountability and transparency in cloud computing environments without sacrificing efficiency.

9. CONCLUSIONS

Here we have addressed the challenge of credential leakage in CP-ABE based cloud storage system by designing an accountable authority and revocable CryptCloud which supports white-box traceability and auditing (referred to as CryptCloud+). This is the first CP-ABE based cloud storage system that simultaneously supports white-box traceability, accountable authority, auditing and effective revocation. Specifically, CryptCloud+ allows us to trace and revoke malicious cloud users (leaking credentials). Our approach can be also used in the case where the users' credentials are redistributed by the semi-trusted authority. We note that we may need black-box traceability, which is a stronger notion (compared to white-box traceability), in CryptCloud. One of our future works is to consider the black-box traceability and auditing. The future of CryptCloud+ lies in continuous innovation. Future research will explore the integration of black-box traceability and auditing mechanisms, further solidifying the security posture of CP-ABE based cloud storage.CryptCloud+ paves the way for a future where cloud storage can be leveraged with greater confidence. By addressing credential leakage and fostering a secure data ecosystem, CryptCloud+ paves the way for a future where sensitive information can be stored and accessed within the cloud with greater peace of mind.

REFERENCES

1. Mazhar Ali, Revathi Dhamotharan, Eraj Khan, Samee U. Khan, Athanasios V. Vasilakos, Keqin Li, and Albert Y. Zomaya. Sedasc: Secure data sharing in clouds. IEEE Systems Journal, 11(2):395–404, 2023.

2. Mazhar Ali, Samee U. Khan, and Athanasios V. Vasilakos. Security in cloud computing: Opportunities and challenges. Inf. Sci., 305:357–383, 2017.

3. Michael Armbrust, Armando Fox, R ean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al. A view of cloud computing. Communications of the ACM, 53(4):50–58, 2015.

4. Nuttapong Attrapadung and Hideki Imai. Attribute-based encryption supporting direct/indirect revocation modes. In Cryptography and Coding, pages 278–300. Springer, 2009.

5. Amos Beimel. Secure schemes for secret sharing and key distribution. PhD thesis, PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

6. Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In Advances in Cryptology-CRYPTO'92, pages 390–420. Springer, 1993.

7. Dan Boneh and Xavier Boyen. Short signatures without random oracles. In EUROCRYPT - 2004, pages 56–73, 2004.

8. Hongming Cai, Boyi Xu, Lihong Jiang, and Athanasios V. Vasilakos. Iot-based big data storage systems in cloud computing: Perspectives and challenges. IEEE Internet of Things Journal, 4(1):75–87, 2017.

9. Jie Chen, Romain Gay, and Hoeteck Wee. Improved dual system ABE in prime-order groups via predicate encodings. In Advances in Cryptology - EUROCRYPT 2015, pages 595–624, 2015.

10. Angelo De Caro and Vincenzo Iovino. jpbc: Java pairing based cryptography. In ISCC 2011, pages 850–855. IEEE, 2011.

11. Hua Deng, Qianhong Wu, Bo Qin, Jian Mao, Xiao Liu, Lei Zhang, and Wenchang Shi. Who is touching my cloud. In Computer Security-ESORICS 2014, pages 362–379. Springer, 2014.

12. Zhangjie Fu, Fengxiao Huang, Xingming Sun, Athanasios Vasilakos, and Ching-Nung Yang. Enabling semantic search based on conceptual graphs over encrypted outsourced data. IEEE Transactions on Services Computing, 2016.

13. Vipul Goyal. Reducing trust in the PKG in identity based cryptosystems. In Advances in Cryptology-CRYPTO 2007, pages 430–447. Springer, 2007.

14. Vipul Goyal, Steve Lu, Amit Sahai, and Brent Waters. Black-box accountable authority identity-based encryption. In Proceedings of the 15th ACM conference on Computer and communications security, pages 427–436. ACM, 2008.

15. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the 13th ACM conference on Computer and communications security, pages 89–98. ACM, 2006.

16. Qi Jing, Athanasios V. Vasilakos, Jiafu Wan, Jingwei Lu, and Dechao Qiu. Security of the internet of things: perspectives and challenges. Wireless Networks, 20(8):2481–2501, 2014.

17. Allison Lewko. Tools for simulating features of Composite order bilinear groups in the prime order setting. In Advances in Cryptology–EUROCRYPT 2012, pages 318–335. Springer, 2012.

18. Allison Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In Advances in Cryptology–EUROCRYPT 2010, pages 62–91. Springer, 2010.

19.Allison Lewko and Brent Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In Advances in Cryptology–CRYPTO 2012, pages 180–198. Springer, 2012.

20. Jiguo Li, Xiaonan Lin, Yichen Zhang, and Jinguang Han. KSFOABE: outsourced attribute-based encryption with keyword search function for cloud storage. IEEE Trans. Services Computing, 10(5):715–725, 2017.