International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

A CRYPTOGRAPHICALLY ENFORCED ACCESS CONTROL WITH A FLEXIBLE USER REVOCATION ON UN-TRUSTED CLOUD

N Ashok [1],Devarasetty Akhila[2], Kavali Bhumika[3], Ch Bharath Reddy[4], K Vamshidhar Reddy[5]

[2,3,4,5] UG Scholars, Department of CSE, **AVN Institute of Engineering and Technology,**Hyderabad, Telangana, India.

[1]Assistant Professor, Department of CSE, **AVN Institute of Engineering and Technology**, Hyderabad, Telangana, India.

ABSTRACT

Enabling cryptographically enforced access controls for data hosted in un trusted cloud is attractive for many users and organizations. However, designing efficient cryptographically enforced dynamic access control system in the cloud is still challenging. In this paper, we propose Crypt-DAC, a system that provides practical cryptographic enforcement of dynamic access control. Crypt-DAC revokes access permissions by delegating the cloud to update encrypted data. In Crypt-DAC, a file is encrypted by a symmetric key list which records a file key and a sequence of revocation keys. In each revocation, a dedicated administrator uploads a new revocation key to the cloud and requests it to encrypt the file with a new layer of encryption and update the encrypted key list accordingly. Crypt-DAC proposes three key techniques to constrain the size of key list and encryption layers. As a result, Crypt-DAC enforces dynamic access control that provides efficiency, as it does not require expensive decryption/re encryption and uploading/re-uploading of large data at the administrator side, and security, as it immediately revokes access permissions. We use formalization framework and system implementation to demonstrate the security and efficiency of our construction.

1. INTRODUCTION

With the considerable advancements in cloud computing, users and organizations re finding it increasingly appealing to store and share data through cloud services. Cloud service providers (such as Amazon, Microsoft, Apple, etc.) Provide abundant cloud based services, ranging from small-scale personal services to large-scale industrial services. However, recent data breaches, such as releases of private photos [10], have raised concerns regarding the privacy of cloud-managed data. Actually, a cloud service provider is usually not secure due to design drawbacks of software and system vulnerability [2], [3]. As such, a critical issue is how to enforce data access control on the potentially untrusted cloud.

In response to these security issues, numerous works [1], [4]–[9] have been proposed to support access control on untrusted cloud services by leveraging cryptographic primitives. Advanced cryptographic primitives are applied for enforcing many access control paradigms. For example, attributebased encryption (ABE) [5] is a cryptographic counterpart of attribute-based access control (ABAC) model [11]. However, previous works mainly consider static scenarios in which access control policies rarely change. The previous works incur high overhead when access control policies need to be changed in practice. At a first glance, the revocation of a user's permission can be done by revoking his access to the keys with which the files are encrypted. This solution, however, is not secure as the user can keep a local copy of the keys before the revocation. To prevent such a problem, files have to be re-encrypted with new keys. This requires the file owner to download the file, re-encrypt the file, and upload it back for the cloud to update the previous encrypted file, incurring prohibitive communication overhead at the file owner side. Currently, only a few works investigated the problem of dynamic data access control. Garrison et al. [12] proposed two revocation schemes. The first scheme requires an administrator to re-encrypt file with new keys as discussed above. This scheme incurs a considerable communication overhead. Instead, the second scheme delegates users to re-encrypt the file when they need to modify the file, relieving the administrator from re-encrypting file data by itself. This scheme, however, comes with a security penalty as the revocation operation is delayed to the next

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

user's modification to the file. As a result, a newly revoked user can still access the file before the next writing operation. Wang et al. [23] proposed another revocation scheme, in which the symmetric homomorphic encryption scheme [24] is used to encrypt the file. Such a design enables the cloud to directly re-encrypt file without decryption. However, this scheme incurs expensive file read/write overhead as the encryption/decryption operation involves comparable overhead with the public key encryption schemes.

To overcome these problems, we present Crypt-DAC, a cryptographically enforced dynamic access control system on untrusted cloud. Crypt-DAC delegates the cloud to update encrypted files in permission revocations. In Crypt-DAC, a file is encrypted by a symmetric key list which records a file key and a sequence of revocation keys. In a revocation, the administrator uploads a new revocation key to the cloud, which encrypts the file with a new layer of encryption and updates the encrypted key list accordingly. Same as previous works [12], [23], we assume a honest-but-curious cloud, i.e., the cloud is honest to perform the required commends (such as re-encryption of files and properly update previous encrypted files) but is curious to passively gathering sensitive information. Although the basic idea of layered encryption is simple, it entails tremendous technical challenges. For instance, the size of key list and encryption layers would increase as the number of revocation operations, which incurs additional decryption overhead for users to access files. To overcome such a problem, Crypt-DAC proposes three key techniques as follows.

First, Crypt-DAC proposes delegation-aware encryption strategy to delegate the cloud to update policy data. For a file, the administrator appends a new revocation key at the end of its key list and requests the cloud to update this key list in the policy data. The size of the key list however increases with the revocation operations, and a user has to download and decrypt a large key list in each file access. To overcome this problem, we adopt the key rotation technique [15] to compactly encrypt the key list in the policy data. As a result, the size of the key list remains constant regardless of revocation operations.

Second, Crypt-DAC proposes adjustable onion encryption strategy to delegate the cloud to update file data. For a file, the administrator requests the cloud to encrypt the file with a new layer of encryption. Similarly, the size of the encryption layers increases with the revocation operations, and a user has to decrypt multiple times in each file access. To overcome this problem, we enable the administrator to define a tolerable bound for the file. Once the size of encryption layers reaches the bound, it can be made to not increase anymore by delegating encryption operations to the cloud. As a result, the administrator can flexibly adjust a tolerable bound for each file (according to file type, access pattern, etc.) To achieve a balance between efficiency and security.

During the life cycle of a file, its encryption layers continuously increase until a pre-defined bound is reached. Crypt- DAC proposes delayed de-onion encryption strategy to periodically refresh the symmetric key list of the file and remove the Bounded encryption layers over it through writing operations. In specific, the next user to write to the file encrypts the writing content by a new symmetric key list only containing a new file key, and updates the key list in the policy data. With this strategy, Crypt-DAC periodically removes the bounded encryption layers of files while amortizing the burden to a large number of writing users.

Altogether, Crypt-DAC achieves efficient revocation, efficient file access and immediate revocation simultaneously. For revocation efficiency, Crypt-DAC incurs lightweight communication overhead at the administrator side as it does not need to download and re-upload file data. For immediate revocation, the permissions of users are immediately revoked as the files are re-encrypted. For file access efficiency, the files are still encrypted by symmetric keys. We have implemented Crypt-DAC as well as several recent works [12], [23] on Alicloud. Real experiments suggest that Crypt-DAC is three orders of magnitude more efficient in communication in access revocation compared with the first scheme in [12], and is nearly two orders of magnitude more efficient in computation in file access compared with the scheme in [23]. Finally, Crypt-DAC is able to immediately revoke access permissions compared with the second scheme in [12].

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

2. SYSTEM ANALYSIS

EXISTING SYSTEM

- Gudes et al. [27] explore cryptography to enforce hierarchy access control without considering dynamic policy scenarios. Akl et al. [28] propose a key assignment scheme to simplify key management in hierarchical access control policy. Also, this work does not consider policy update issues. Later, Atallah et al. [29] propose a method that allows policy updates, but in the case of revocation, all descendants of the affected node in the access hierarchy must be updated, which involves high computation and communication overhead.
- Ibraimi et al. [30] cryptographically support role based access control structure using mediated public encryption. However, their revocation operation relies on additional trusted infrastructure and an active entity to re-encrypt all affected files under the new policy. Similarly, Nali et al. [31] enforce role based access control structure using public-key cryptography, but requires a series of active security mediators. Ferrara et al. [32] define a secure model to formally prove the security of a cryptographically enforced RBAC system. They further show that an ABE-based construction is secure under such model. However, their work focuses on theoretical analysis.
- Pirretti et al. [33] propose an optimized ABE-based access control for distributed file systems and social networks, but their construction does not explicitly address the dynamic revocation. Sieve [23] is a attribute based access control system that allows users to selectively expose their private data to third web services. Sieve uses ABE to enforce attribute based access policies and homomorphic symmetric encryption [24] to encrypt data. With homomorphic symmetric encryption, a data owner can delegate revocation tasks to the cloud assured that the privacy of the data is preserved. This work however incurs prohibitive computation overhead since it adopts the homomorphic symmetric encryption to encrypt files.
- GORAM [25] allows a data owner to enforce an access matrix for a list of authorized users and provides strong data privacy in two folds. First, user access patterns are hidden from the cloud by using ORAM techniques [26]. Second, policy attributes are hidden from the cloud by using attribute-hiding predicate encryption [21], [22]. The cryptographic algorithms, however, incur additional performance overhead in data communication, encryption and decryption. Also, GORAM does not support dynamic policy update. Over encryption [34], [35] is a cryptographical method to enforce an access matrix on outsourced data. Over-encryption uses double encryption to enforce the whole access matrix. As a result, the administrator has to rely on the cloud to run complex algorithms over the matrix to update access policy, assuming a high level of trust on the cloud.

PROPOSED SYSTEM

- The proposed system presents Crypt-DAC, a cryptographically enforced dynamic access control system on un trusted cloud. Crypt-DAC delegates the cloud to update encrypted files in permission revocations. In Crypt-DAC, a file is encrypted by a symmetric key list which records a file key and a sequence of revocation keys. In a revocation, the administrator uploads a new revocation key to the cloud, which encrypts the file with a new layer of encryption and updates the encrypted key list accordingly. Same as previous works, we assume a honest-but-curious cloud, i.e., the cloud is honest to perform the required commends (such as re-encryption of files and properly update previous encrypted **files**) but is curious to passively gathering sensitive information. Although the basic idea of layered encryption is simple, it entails tremendous technical challenges. For instance, the size of key list and encryption layers would increase as the number of revocation operations, which

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

incurs additional decryption overhead for users to access files. To overcome such a problem, Crypt-DAC proposes three key techniques as follows.

- First, Crypt-DAC proposes delegation-aware encryption strategy to delegate the cloud to update policy data. For a file, the administrator appends a new revocation key at the end of its key list and requests the cloud to update this key list in the policy data. The size of the key list however increases with the revocation operations, and a user has to download and decrypt a large key list in each file access. To overcome this problem, we adopt the key rotation technique to compactly encrypt the key list in the policy data. As a result, the size of the key list remains constant regardless of revocation operations.

- Second, Crypt-DAC proposes adjustable onion encryption strategy to delegate the cloud to update file data. For a file, the administrator requests the cloud to encrypt the file with a new layer of encryption. Similarly, the size of the encryption layers increases with the revocation operations, and a user has to decrypt multiple times in each file access. To overcome this problem, we enable the administrator to define a tolerable bound for the file. Once the size of encryption layers reaches the bound, it can be made to not increase anymore by delegating encryption operations to the cloud. As a result, the administrator can flexibly adjust a tolerable bound for each file (according to file type, access pattern, etc.) To achieve a balance between efficiency and security.
- Crypt- DAC proposes delayed de-onion encryption strategy to periodically refresh the symmetric key list of the file and remove the bounded encryption layers over it through writing operations. In specific, the next user to write to the file encrypts the writing content by a new symmetric key list only containing a new file key, and updates the key list in the policy data. With this strategy, Crypt-DAC periodically removes the bounded encryption layers of files while amortizing the burden to a large number of writing users.

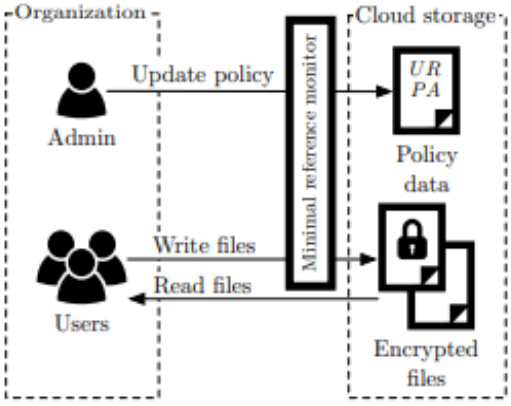


Fig. 1: Diagram of a cloud storage system

MODULES:

- Data Owner
- Data User
- Semi-trusted authority
- Auditor
- Cloud Server and Encryption Module

EXISTING SYSTEM:

- In a CP-ABE based cloud storage system, for example, organizations (e.g., a university such as the University of Texas at San Antonio) and individuals (e.g., students, faculty members and visiting scholars of the university) can first specify access policy over attributes of a potential cloud user.
- Authorized cloud users then are granted access credentials (i.e., decryption keys) corresponding to their attribute sets (e.g., student role, faculty member role, or visitor role), which can be used to obtain access to the outsourced data.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

- As a robust one-to-many encryption mechanism, CP-ABE offers a reliablemethod to protect data stored in cloud,but also enables fine-grained accesscontrol over the data.

DISADVANTAGES OF EXISTINGSYSTEM:

- The leakage of any sensitive student information stored in cloud could resultin a range of consequences for theorganization and individuals (e.g., litigation, loss of competitive advantage, and criminal charges).
- The existing CP-ABE based cloudstorage systems fail to consider the casewhere access credential is misused.

PROPOSED SYSTEM:

- Seeking to mitigate access credential misuse, we propose CryptCloud+, an accountable authority and revocable CPABE based cloud storage system with white-box traceability and auditing.
- Specifically, in our work, we first presenta CP-ABE based cloud storage framework. Using this (generic) framework, we propose two accountable authority and revocable CP-ABE systems (with whitebox traceability and auditing) that are fully secure in the standard model, referred to as ATER-CP-ABE and ATIR-CPABE, respectively. Based on the two systems, we present the construction of CryptCloud+
- Access credentials for individual tracedand further determined to be "compromised" can be revoked.

ADVANTAGES OF PROPOSEDSYSTEM:

- To the best of our knowledge, this is the first practical solution to secure fine- grained access control over encrypted data in cloud.
- Users who leak their access credentials can be traced and identified.
- A semi-trusted authority, who (without proper authorization) generates and further distributes access credentials to unauthorized user(s), can be identified.

This allows further actions to beundertaken (e.g. criminal investigation orcivil litigation for damages and breach of contract).

- An auditor can determine if a (suspected) cloud user is guilty in leaking his/her access credential.

MODULES DESCSRIPTION:

Data Owner:

In the first module, we develop the Data Owner Module. In this module, data owner has the option of File Upload, File View, Trace Request and Trace Results. This module helps the owner to register those details and also include login details. This module helps the owner to upload his file with encryption algorithm. This ensures the files to be protected from unauthorized user. Data owner has a collection of documentsthat he wants to outsource to the cloud server in encrypted form while still keeping the capability to search on them for effective utilization. Data Owners (DOs) encrypt their data under therelevant access policies prior to outsourcing the (encrypted)data to a public cloud (PC).PC stores the outsourced (encrypted) data from Dosand handles data access requests from data users(DUs)

Data User:

This module includes the user registration login details.This module is used to help the client to search the file using the multiple key words concept and get the accurate result list based on the user query. The user is going to select the required file and register the user details and get activation code in mail email before enter the activation code. After user can download the Zip file and extract that file.Data users are authorized ones to access the documentsof data owner. With *t* query keywords, theauthorized user can generate a trapdoor *TD* accordingto search control mechanisms to fetch *k* encrypted documentsfrom cloud server. Then, the data user can decryptthe documents with the

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

shared secret key.Authorized DUs are able to access (e.g. downloadand decrypt) the outsourced data.

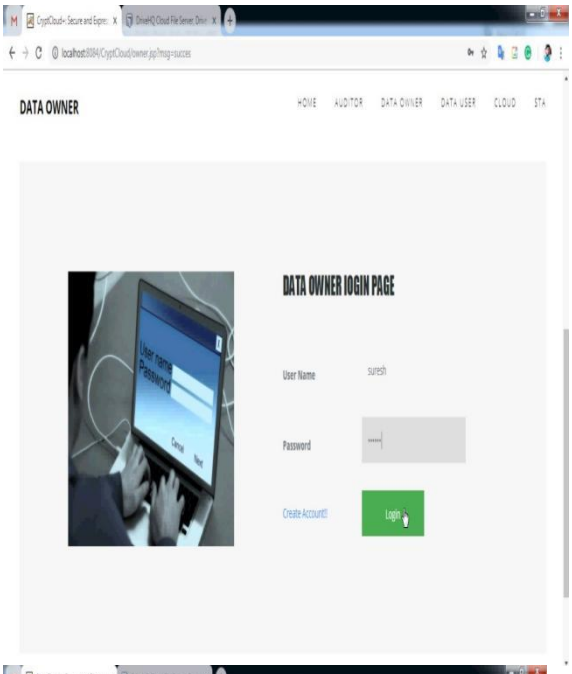
Semi-trusted authority:

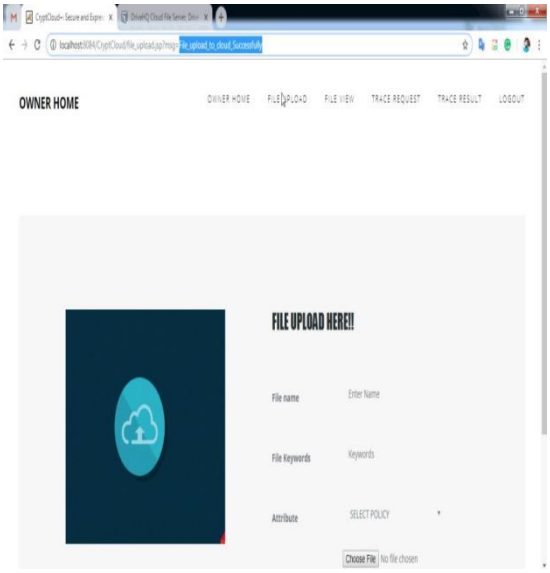
Semi-trusted authority (STA)generates system parametersand issues access credentials (i.e., decryptionkeys) to DUs.

Auditor:

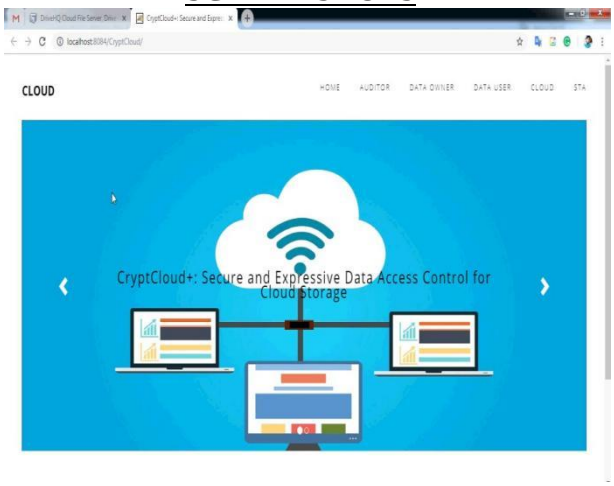
Auditor (AU) is trusted by other entities, takescharge of audit and revoke procedures, and retursthe trace and audit results to DOs and DUs. In this module, auditor has the options of File details, User Request & Trace Request details.

Cloud Server and Encryption Module: This module is used to help the server to encrypt the document using RSA Algorithmand to convert the encrypted document to the Zip file with activation code and thenactivationcode send to the user for download.Cloud serverstores the encrypted document collectionfor dataowner. Uponreceiving the trapdoor *TD* from the datauser, the cloud server executes search, and finally returns the corresponding collection of top-*k*ranked encrypted documents. Besides, upon receivingthe update information from the data owner, the serverneeds to update and document collection *C*according to the received information.The cloud server in the proposed scheme is consideredas "honest-but-curious", which is employed by lots ofworks on secure cloud data search





SCREEN SHOTS



International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

CONCLUSION

We presented Crypt-DAC, a system that provides practical cryptographic enforcement of dynamic access control in the potentially untrusted cloud provider. Crypt-DAC meets its goals using three techniques. In particular, we propose to delegate the cloud to update the policy data in a privacy-preserving manner using a delegation-aware encryption strategy. We propose to avoid the expensive re-encryptions of file data at the administrator side using a adjustable onion encryption strategy. In addition, we propose a delayed de-onion encryption strategy to avoid the file reading overhead. The theoretical analysis and the performance evaluation show that Crypt-DAC achieves orders of magnitude higher efficiency in access revocations while ensuring the same security properties under the honest but curious threat model compared with previous schemes.

REFERENCES

[1] J. Bethencourt, A. Sahai, and B. Waters, Ciphertext-policy attribute based encryption, in IEEE S&P, 2007.

[2] X. Wang, Y. Qi, and Z. Wang, Design and Implementation of secpod: A Framework for Virtualization-based Security Systems, IEEE

Transactions on Dependable and Secure Computing, vol. 16, no. 1, 2019.

[3] J. Ren, Y. Qi, Y. Dai, X. Wang, and Y. Shi, appsec: A Safe Execution Environment for Security SensitiveApplications, in ACM VEE, 2015.

[4] V. Goyal, A. Jain, O. Pandey, and A. Sahai, Bounded ciphertext policy attribute based encryption,in ICALP, 2008.

[5] V. Goyal, O. Pandey, A. Sahai, and B.Waters, Attribute-based encryption for fine-grained access control of encrypted data, in ACM CCS, 2006.

[6] J. Katz, A. Sahai, and B. Waters, Predicate encryption supporting disjunctions, polynomial equations, and inner products, in EUROCRYPT, 2008.

[7] S. Muller and S. Katzenbeisser, Hiding the policy in cryptographic access control, in STM, 2011.

[8] R. Ostrovsky, A. Sahai, and B. Waters, Attribute- based encryption with non-monotonic access structures, in ACM CCS, 2007.

[9] A. Sahai, and B. Waters, Fuzzy identity-based encryption, in EUROCRYPT, 2005.

[10] T. Ring, Cloud computing hit by celebgate, http://www.scmagazineuk. Com/cloud-computing-hit-by-celebgate/article/370815/, 2015.

[11] X. Jin, R. Krishnan, and R. S. Sandhu, A unified attribute-based access control model covering DAC, MAC and RBAC, in ddbsec, 2012.

[12] W. C. Garrison III, A. Shull, S. Myers, and, A. J. Lee, On the Practicality of Cryptographically Enforcing Dynamic Access Control Policies in the Cloud, in IEEE S&P, 2016.

[13] R. S. Sandhu, Rationale for the RBAC96 family of access control models, in proceedings of ACM Workshop on RBAC, 1995.

[14] T. Jiang, X. Chen, Q. Wu, J. Ma, W. Susilo, and W. Lou, Secure and Efficient Cloud Data Deduplication With Randomized Tag, IEEE Trasactions on Information Forensics and Security, vol. 12, no. 3, 2017.

[15] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, K. Fu, Plutus: Scalable Secure File Sharing on Untrusted Storage, in proceedings of USENIX FAST, 2003.