Crypt-DAC: REVOKES ACCESS PERMISSIONS BY DELEGATING THE CLOUD TO UPDATE ENCRYPTED DATA

Mrs. K. Anitha, M.C.A, M.Tech (CSE), Head, Department of Computer Science, Sri Durga Malleswara Siddhartha Mahila Kalasala, Vijayawada

ABSTRACT:

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.

KEYWORDS: Crypt-DAC , Dynamic Access Control, Encryption And Uploading.

INTRODUCTION

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 cloud computing, users and organizations are 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, have raised concerns regarding the privacy of cloudmanaged data. Actually, a cloud service provider is usually not secure due to design drawbacks of software and system vulnerability. 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 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, attribute-based encryption (ABE) is a cryptographic

DOI : 10.36893.DRSR.2021.V08I14.11-17

counterpart of attribute-based access control (ABAC) model. 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. 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 reencrypting file data by itself. This scheme, however, comes with a security penalty as the revocation operation is delayed to the next 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. proposed another revocation scheme, in which the symmetric homomorphic encryption scheme 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.

PROBLEM DEFINTION

- 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

DOI : 10.36893.DRSR.2021.V08I14.11-17

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.

Disadvantages

- In the existing work, the system doesn't have mote security due to lack of Delegation-aware encryption and Adjustable onion encryption.
- The system proposes a key assignment scheme to simplify key management in hierarchical access control policy. Also, this work does not consider policy update issues.

LITERATURE SURVEY

- **Gudes et al.** 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.
- **Atallah et al.** 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.** 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.
- **Nali et al.** enforce role based access control structure using public-key cryptography, but requires a series of active security mediators.
- **Ferrara et al.** 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.

PROPOSED APPROACH

- 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

DOI : 10.36893.DRSR.2021.V08I14.11-17

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 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.

DOI : 10.36893.DRSR.2021.V08I14.11-17

Advantages

- Crypt-DAC achieves efficient revocation, efficient file access and immediate revocation simultaneously.
- The system stores encrypted data on the cloud, but never reveals the decryption keys to the cloud. This protects the confidentiality of the file data

IMPLEMENTATION

System Model:

Data Owner (Alice): In this module we executed by the data owner to setup an account on an untrusted server. On input a security level parameter 1^{λ} and the number of ciphertext classes n (i.e., class index should be an integer bounded by 1 and n), it outputs the public system parameter param, which is omitted from the input of the other algorithms for brevity.

Network Storage: With our solution, Alice can simply send Bob a single aggregate key via a secure e-mail. Bob can download the encrypted photos from Alice's Dropbox space and then use this aggregate key to decrypt these encrypted photos. In this Network Storage is untrusted third party server.

Key Generation

Public-key cryptography, also known as asymmetric cryptography, is a class of cryptographic algorithms which requires two separate keys, one of which is secret (or private) and one of which is public. Although different, the two parts of this key pair are mathematically linked.

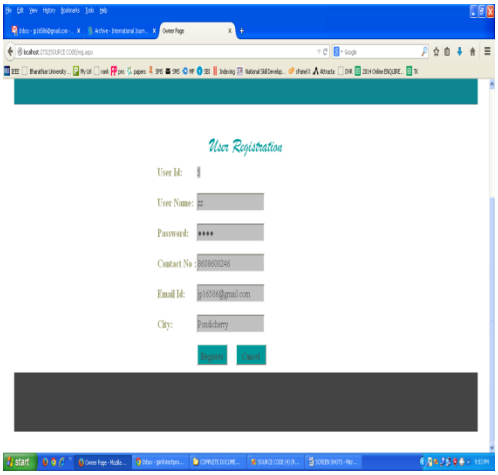
The public key is used to encrypt plaintext whereas the private key is used to decrypt ciphertext .Data owner to randomly generate a public/master-secret key pair.

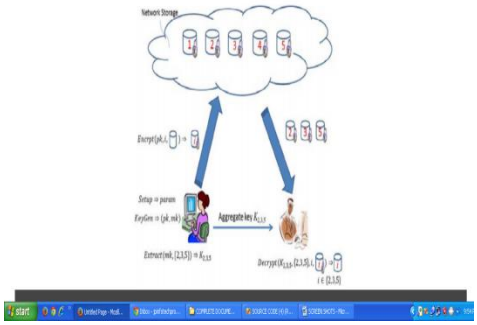
Encryption

Encryption keys also come with two flavours symmetric key or asymmetric (public) key. Using symmetric encryption, when Alice wants the data to be originated from a third party, she has to give the encrypted her secret key; obviously, this is not always desirable.

By contrast, the encryption key and decryption key are different in public key encryption. The use of public-key encryption gives more flexibility for our applications.

DOI : 10.36893.DRSR.2021.V08I14.11-17





CONCLUSION

In this paper, presented Cryptographically enforced access controls , a system that provides practical cryptographic enforcement of dynamic access control in the potentially untrusted cloud provider. Cryptographically enforced access controls 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 an adjustable onion encryption strategy. The theoretical analysis and the performance evaluation show that Crypt- DAC achieves orders of magnitude higher efficiency.

REFERENCES:

[1] S.S.M. Chow, Y.J. He, L.C.K. Hui, and S.-M. Yiu, “SPICE – Simple Privacy-Preserving Identity-Management for Cloud Environment,” Proc. 10th Int’l Conf. Applied Cryptography and Network Security (ACNS), vol. 7341, pp. 526-543, 2012.

[2] L. Hardesty, Secure Computers Aren’t so Secure. MIT press, http:// www.physorg.com/news176107396.html, 2009.

[3] C. Wang, S.S.M. Chow, Q. Wang, K. Ren, and W. Lou, “Privacy-Preserving Public Auditing for Secure Cloud Storage,” IEEE Trans. Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.

DOI : 10.36893.DRSR.2021.V08I14.11-17

[4] B. Wang, S.S.M. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," Proc. IEEE 33rd Int'l Conf. Distributed Computing Systems (ICDCS), 2013.

[5] S.S.M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R.H. Deng, "Dynamic Secure Cloud Storage with Provenance," Cryptography and Security, pp. 442-464, Springer, 2012.

[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," Proc. 22nd Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT '03), pp. 416-432, 2003.

[7] M.J. Atallah, M. Blanton, N. Fazio, and K.B. Frikken, "Dynamic and Efficient Key Management for Access Hierarchies," ACM Trans. Information and System Security, vol. 12, no. 3, pp. 18:1-18:43, 2009.

[8] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records," Proc. ACM Workshop Cloud Computing Security (CCSW '09), pp. 103-114, 2009.

[9] F. Guo, Y. Mu, Z. Chen, and L. Xu, "Multi-Identity Single-Key Decryption without Random Oracles," Proc. Information Security and Cryptology (Inscrypt '07), vol. 4990, pp. 384-398, 2007.

[10] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," Proc. 13th ACM Conf. Computer and Comm. Security (CCS '06), pp. 89-98, 2006.

[11] S.G. Akl and P.D. Taylor, "Cryptographic Solution to a Problem of Access Control in a Hierarchy," ACM Trans. Computer Systems, vol. 1, no. 3, pp. 239-248, 1983.

[12] G.C. Chick and S.E. Tavares, "Flexible Access Control with Master Keys," Proc. Advances in Cryptology (CRYPTO '89), vol. 435, pp. 316-322, 1989.

[13] W.-G. Tzeng, "A Time-Bound Cryptographic Key Assignment Scheme for Access Control in a Hierarchy," IEEE Trans. Knowledge and Data Eng., vol. 14, no. 1, pp. 182-188, Jan./Feb. 2002.

[14] G. Ateniese, A.D. Santis, A.L. Ferrara, and B. Masucci, "Provably-Secure Time-Bound Hierarchical Key Assignment Schemes," J. Cryptology, vol. 25, no. 2, pp. 243-270, 2012.

DOI : 10.36893.DRSR.2021.V08I14.11-17