INFORMATYKA I MATEMATYKA

**Amitava Nag, Arup Kumar Chattopadhyay,
Koustav Chanda, Shayak Sadhu**

ZASTOSOWANIE KRYPTOGRAFII DLA ZAPEWNIENIA BEZPIECZEŃSTWA I PRYWATNOŚCI DANYCH WRAŻLIWYCH W CHMURZE

[**słowa kluczowe**: bezpieczeństwo, prywatność, kryptografia, przechowywanie danych w chmurze]

Streszczenie

Przechowywanie danych w chmurze jest ważną aplikacją chmury obliczeniowej, która jest udostępniana organizacjom. Jednak problemy związane z prywatnością i bezpieczeństwem wrażliwych danych w chmurze stanowią istotne wyzwanie dla właścicieli danych. Aby zapewnić prywatność i bezpieczeństwo danych, większość istniejących programów proponuje przekazywanie wyłącznie zaszyfrowanych danych do zasobów Dostawcy Usług Chmurowych (CSP). W niniejszym dokumencie proponujemy model bezpieczeństwa dla systemów przechowywania danych w chmurze, który zapewnia bezpieczeństwo tychże danych. W proponowanym schemacie poufność danych w chmurze jest obsługiwana przez szyfrowanie symetryczne, a bezpieczeństwo podczas dzielenia danych zapewnia szyfrowanie klucza publicznego. Analiza pokazuje, że proponowany schemat zapewnia akceptowalny poziom bezpieczeństwa wrażliwych danych w chmurze.

* * *

Amitava Nag, Arup Kumar Chattopadhyay,
Koustav Chanda, Shayak Sadhu

SECURITY AND PRIVACY OF SENSITIVE DATA IN CLOUD STORAGE THROUGH CRYPTOGRAPHIC APPROACH

[**key words**: security, privacy, cryptography, cloud storage]

Abstract

Cloud storage is an important application of cloud computing that offers organizations to store data in the cloud. However, privacy and security problems of sensitive data on cloud are the major challenges for data owners. To ensure data privacy and security, most of existing schemes propose to outsource only encrypted data to the cloud storage of a Cloud Service Provider (CSP). In this paper, we propose a security model for cloud storage systems which ensures the data security in the cloud. In the proposed scheme, confidentiality of data in the cloud is handled by symmetric encryption and security during sharing of data is ensured by public key encryption. The analysis shows that proposed scheme provides acceptable level of security for the sensitive data in cloud storage.

Introduction

Cloud computing is an emerging field in computer science that offers utilization of computing resources through the Internet [1]. Cloud Computing abstracts the physical machine from end users. Upgrading or downgrading of resources is possible in accordance to the plan chosen. Data storage service is one of the most important and popular application of cloud computing. Data storage services abstracts the way of data storage from the user and comes with elastic storage. Elastic storage facilitates on demand expanding or reducing the data storage capacity. Nowadays, many organizations have shifted from utilization of in-house data storage system to cloud storage. Data owner can easily upload personal or confidential data in cloud storage of a Cloud Service Provider (CSP). Dropbox (www.dropbox.com), Google Drive (http://drive.google.com), and Mozy (http://mozy.com) are some popular cloud storage service providers where a data owner can store their data with free of cost to a certain limit. However, privacy and security problems of sensitive information on cloud are still a matter of main concern as data owners do not have direct control over their data [2, 3]. Deyan Chen et al. [14] have shown the current concerns regarding data security in cloud as they are third party services. To protect sensitive data from intrusions and attacks,

some cryptosystems have been proposed for cloud computing [4, 5]. Encryption is a promising cryptographic solution where before being stored in the cloud, data can be encrypted by data owners [6]. Akashdeep Bhardwaj et al. [13] have shown the growing concerns regarding security in cloud. They have illustrated some cryptographic algorithm that can be used for data and link encryption. In a cryptographic system, to ensure the data security in the cloud, the other processes such as key management, access control are also maintained by the data owners. However, to protect confidentiality of sensitive data during sharing among a set of users, some simple but robust cryptographic solutions are needed [7, 8]. Shweta Kaushik et al. [12] have proposed a hybrid symmetric key encryption of data in cloud. They have used symmetric key for data encryption as it is more secure and faster than public key cryptography. The authors in [9] proposed attribute based encryption (ABE) to store encrypted data in the cloud. Mollah et al. [10] design a lightweight cryptographic scheme for cloud-assisted IoT where all data are stored in encrypted form. Although both the schemes supporting secure data access control, but these schemes cannot support forward access control (accessing future data by a departing user) and backward access control (accessing past data by a new user) [11]. Ali et al. [11] proposed a simple but effective secure data sharing scheme for cloud storage which solves forward access control as well as backward access control. Nevertheless, Ali et al. does not check whether a requesting user is authorized or not. By considering the aforementioned limitations we propose a security scheme for cloud storage services that supports secure data access control including forward as well as backward access control and offers user authentication. The major contributions of the proposed work are summarized as follows:

- First, the proposed work ensures the privacy of the sensitive data in the cloud with the help of secret key encryption;
- Next, as in this scheme, the secret key is shared among data owner and data users with the help of public key encryption, thereby, greatly preventing attacks from cyber-criminals around the world;
- Then, all security operations are handled by a cryptographic server (CS) which is a trusted party, thereby, data is secured from forward and backward access control;
- Finally, the data owners, data users and CSP can communicate with each other through a secure channel.

Analysis of the proposed scheme demonstrates that it is secure, effective, and efficient solution.

Proposed Scheme

In this section, we discuss our proposed security solution for data sharing in cloud storage. The outline of our system model is shown in Fig 1.

System Design

Our proposed scheme works with four types of entities as follows:

- Cloud Service Provider (CSP): CSP is usually a commercial organization that provides cloud storage services and behaves as a semi-trusted party because CSP is curious about the contents of hosted data.
- Cryptographic Server (CS): CS is a trusted third party, e.g. government authority. It handles encryption, decryption, key management and access control.
- Data Owner (W): Data owner is an ordinary client of cloud storage who stores its data in the cloud. Data owner is only responsible to generate access control list (ACL) for a particular data file and submit it to the CS. ACL contains a list of users.
- Data Users (U): Data users are the clients who are authorized to access the data hosted by data owner.

Required Keys

In the proposed model, for secure communication among the data owner, the data users and the CSP, several keys are needed. For encryption of data, the CS provides a symmetric key known as cryptographic or secret key (K). The data owner W and the data user U both maintain a pair of keys (PU_W, PR_W) and (PU_U, PR_U) respectively, where PU denotes public key and PR denotes private key.

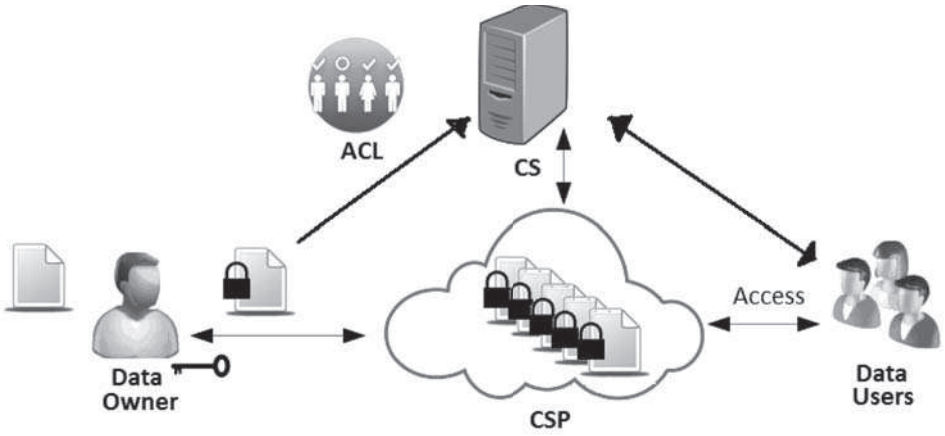

Fig 1: The Basic Principle of the secure data sharing in cloud storage

In the proposed scheme, data owner W submits ACL to CS and requests CS for the cryptographic key (K) to encrypt her file. The CS generates a symmetric key as the cryptographic key for W and sends it after encrypting it with the public key (PU_W) of W. To retrieve the cryptographic key, W decrypts it width her private key (PR_W) and encrypts her file by the retrieved cryptographic key. Subsequently, the encrypted file is uploaded to the CSP for storage by the data owner. For more security, CS destroys the cryptographic key. Before destroying, for each data user U, CS splits the key into two parts in such a way that the key cannot be regenerated by a single part. One part of the key is shared with the corresponding data user after encrypting by the public key of U. On the other hand, CS keeps the other part within the ACL submitted by W. If any user wishes to access the file, she first with the help of her private key decrypts the portion of the key sent by CS and submits it to the CS with a download request. Then the CS regenerates the original cryptographic key by the user portion and the portion which is kept in the ACL. Subsequently, the data file is downloaded from the CSP and decrypted by the cryptographic key provided by CS. The original data file is then sent to the user to access.

Proposed Algorithms

We assume the following standard functions are used for our proposed scheme:

- *PKE*: A standard public key encryption function
- *PKD*: A standard public key decryption function, corresponds to *PKE*
- *SKE*: A secret key encryption function
- *SKD*: A secret key decryption function corresponds, to *SKE*
- H(■): A one-way collision resistance hash function

Key sharing between data owner W and CS. For each upload requested by the data owner W, the CS generates a unique cryptographic key K. W first uploads the ACL for the data file, df in CS and requests the secret key for uploading the data file. Then, CS encrypts the K using the public key of W, PU_W and send it to W on public channel (as presented in Algorithm 1). W can retrieve the cryptographic key K using her private key PR_W (as presented in Algorithm 2).

Key sharing between user U and CS. For each user U available in ACL for data file df, CS generates a random number R_U and computes a key share K_U as $K_U := K \oplus R_U$. Now, CS encrypts the key share with the public key of U, PU_U and transmits it to U (as represented in Algorithm 1), such that only U can retrieve it with its private key PR_U (as represented in Algorithm 2).

CS destroys the secret key. CS stores the R_U with the entry for U in ACL of the data file df. To ensure secrecy CS destroy the secret key K, by overwriting technique.

Algorithm 1: Key management
// Input: *ACL, K,* PU_W, PU_U, *PKE*

1. CS encrypts K as $CK_W := PKE\ (K,\ PU_W)$
2. CS sends CK_W to W
3. For each U in the ACL, CS do the following
4. generates a random number R_U
5. Computes $K_U := K \oplus R_U$
6. Encrypts K_U as $CK_U := PKE\ (K_U,\ PU_U)$
7. sends CK_U to U
8. Delete K and K_U

Algorithm 2: Key retrieve
// Input: CK_W, CK_U, PR_W, PR_U, *PKD*

1. W retrieve K as $K := PKD\ (CK_W,\ PR_W)$
2. U retrieve K_U as $K_U := (CK_U,\ PR_U)$

Uploading of encrypted data file in CSP. The data owner W encrypts the data file df and upload it to CSP (as represented in Algorithm 3). CS also computes the hash code h_s as $h_s = H(cf)$ and stores the h_s in ACL for df.

Algorithm 3: Data upload
// Input: df, *K, SKE*

1. W encrypts df with K as $cf = SKE(df, K)$
2. Upload cf in cloud storage

Download the data file from CSP. The data user U requests to download data file df along with the shared key K_U. First, CS verifies if U belongs to the ACL of data file df. CS generates the hash h'_s for the encrypted version of the file cf and verifies $h'_s = h_s$ (h_s is available at ACL for the given file df) to ensure the integrity (file is not corrupted or modified)) of the file. The entry of U in ACL provides another portion of the key R_U. Then, CS constructs the secret key K by $K = R_U \oplus K_U$. Hence, it decrypts the encrypted data file, transmits the data file to U (as represented in Algorithm 4.).

Algorithm 4: Data download
//Input: cf, *ACL, SKD*

1. CS gets K_U from U
2. Check whether R_U exists in the ACL, if yes then retrieve
3. $K := R_U \oplus K_U$
4. $df := SKD(cf, K)$
5. Send df to U for access

Note that CS gets cf either from U or downloads it from the cloud storage, CSP.

Comparison and Discussion

Table 1 compares the proposed scheme with that of work presented elsewhere. It shows that the proposed scheme provides following services:

- Confidentiality: Data in the cloud storage are encrypted with a secret key, which is kept only with the data owner as after use CS deletes (step 8 Algorithm 1). Thus there is no chance of data leakage until intruder gets the secret key.
- Use of cryptographic server (CS): Cryptographic server is trusted third party that handles Key management, encryption and decryption.

- Integrity of data file: The integrity of the data file is ensured by the hash function used in CS and the hash value stored at ACL.
- Forward and backward access control: Data is safe from inside attacker as uses don‘t have the access to the secret key.
- ACL based access policy: More secured user verification done by CS as only valid Data User with their part of the secret key can only retrieve the data.

The proposed scheme takes advantage of ACL to add more security to symmetric key for encrypted data in cloud. Ali et al. [11] have maintained ACL for maintaining full authorization of the user and could pose a problem when someone duplicates this list. In our scheme we generate ACL in such a way that only partial keys are is present in ACL and the rest of the portion of the key is with the Data User. This ensures security at both ends.

Table 1. Comparison.

	Yan et al.[9]	Mollah et al.[10]	Ali et al.[11]	Proposed
Confidentiality	YES	YES	YES	YES
Cryptographic Server	NO	NO	YES	YES
Forward Access Control	NO	NO	YES	YES
Backward Access Control	NO	NO	YES	YES
Access Policy	Reencryption key generation	Reencryption key generation	ACL	ACL

Conclusion

Recently, the cloud is a big blessing as people can store huge amount of data such as music, messages, photos and so on in the cloud storage at little or no cost. But data security and access control are the few of the most challenging issues. In this paper, we have proposed an access control list (ACL) based security model for cloud storage systems. The main objective of this work is to securely store and access customer’s sensitive data in the cloud. The performance of the proposed work is analyzed and proved that this scheme is efficient and can be used in secure data outsourcing in the cloud.

References:

1. Sadiku, M. N. O. Musa, S. M. and Momoh, O.D. :Cloud Computing: Opportunities and Challenges, *IEEE Potentials*,33(1),34-36 (2014).
2. T. T. Wu, W. C. Dou, C. H. Hu and J. J. Chen, "Service mining for trusted service composition in cross-cloud environment," IEEE Systems Syst. J., vol. PP, no. 99, pp. 1-12, 2014.
3. M. Ali, S. U. Khan and A. V. Vasilakos, "Security in cloud computing: Opportunities and challenges," Inf. Sci., vol. 305, pp. 357-383, 2015.
4. L. Wei, H. Zhu, Z. Cao, Y. Chen and A. V. Vasilakos, "Security and privacy for storage and computation in cloud computing," *Inf. Sci.*, vol. 258,pp. 371-386, Feb. 2014.
5. Cloud security Alliance, "Security guidelines for critical areas of focus in cloud computing v3.0," 2011.
6. Laurence T. Yang, Gaoyuan Huang, Jun Feng, Li Xu. "Parallel GNFS algorithm integrated with parallel block Wiedemann algorithm for RSA security in cloud", Information Sciences, Volume 387, May 2017, Pages 254-265.
7. Mollah, Muhammad Baqer, MdAbulKalam Azad, and Athanasios Vasilakos. "Security and privacy challenges in mobile cloud computing: Survey and way ahead." *Journal of Network and Computer Applications* (2017).
8. Ali, Mazhar, Samee U. Khan and Athanasios V. Vasilakos. "Security in cloud computing: Opportunities and challenges." *Information Sciences* 305 (2015): 357-383.
9. Z. Yan, M. Wang, Y. Li and A. V. Vasilakos, "Encrypted Data Management with Deduplication in Cloud Computing," in *IEEE Cloud Computing*, vol. 3, no. 2, pp. 28-35, Mar.-Apr. 2016
10. M. B. Mollah, M. A. K. Azad and A. Vasilakos, "Secure Data Sharing and Searching at the Edge of Cloud-Assisted Internet of Things," in *IEEE Cloud Computing*, vol. 4, no. 1, pp. 34-42, Jan.-Feb. 2017.
11. M. Ali *et al*., "SeDaSC: Secure Data Sharing in Clouds," in *IEEE Systems Journal*, vol. 11, no. 2, pp. 395-404, June 2017.
12. Shweta Kaushik and Charu Gandhi. "Cloud data security with hybrid symmetric encryption." In Computational Techniques in Information and Communication Technologies (ICCTICT), 2016 International Conference on, pp. 636-640. IEEE, 2016.
13. Akashdeep Bhardwaj, G. V. B. Subrahmanyam, Vinay Avasthi, and HanumatSastry. "Security Algorithms for Cloud Computing." Procedia Computer Science 85 (2016): 535-542.
14. Deyan Chen, and Hong Zhao. "Data security and privacy protection issues in cloud computing." In Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on, vol. 1, pp. 647-651. IEEE, 2012.