Data Storage Security Model for Cloud Computing

Hiren B. Patel[1], Dhiren R. Patel[2], Bhavesh Borisaniya[2], and Avi Patel[3]

[1] S.P. College of Engineering, Visnagar, India
[2] S.V. National Institute of Technology, Surat, India
[3] City University, London, UK
{hbpatel1976,dhiren29p,borisaniyabhavesh,avi2687}@gmail.com

Abstract. Data security is one of the biggest concerns in adopting Cloud computing. In Cloud environment, users remotely store their data and relieve themselves from the hassle of local storage and maintenance. However, in this process, they lose control over their data. Existing approaches do not take all the facets into consideration viz. dynamic nature of Cloud, computation & communication overhead etc. In this paper, we propose a Data Storage Security Model to achieve storage correctness incorporating Cloud's dynamic nature while maintaining low computation and communication cost.

Keywords: Cloud Computing, Data Storage Correctness, Privacy, Security.

1 Introduction

The apparent benefit of having Cloud computing model is to relax the user from the lumber of storing and maintaining data or computing resources locally. This reduces the initial investment of any organization drastically, and provides a pay-as-you-go model. In spite of these noticeable advantages, Cloud computing has not been adopted widely in practice due to security and privacy concerns. Along with these, other traditional IT security issues such as integrity, confidentiality, availability, reliability, non-repudiation, efficient retrieval, data sharing etc. have the same significance in Cloud computing. Among all these, data storage correctness is one of the important security issues in Cloud.

There are various methods being adopted for the data storage correctness. Trusted third party such as cryptographic coprocessors is preferred by many researchers [2] [3] [6] [8] [11]. It adds additional cost on Cloud users' part for extra hardware. One implement the functionalities of cryptographic coprocessor using open source code in form of client application [1] [8]. It can be proved as cost-effective solution with some compromise at performance level.

In this paper, we aim to provide a client application based Data Storage Security Model. Rest of the paper is organized as follows. Section 2 discusses the recent work carried out followed by problem statement in Section 3. Section 4 presents our proposed scheme in detail along with the validation the planned-goals with the design. Section 5 includes some possible techniques to implement the core components of this model. With conclusions in section 6 follows references at the end.

V.V. Das and J. Stephen (Eds.): CNC 2012, LNICST 108, pp. 37–45, 2012.

2 Related Work

This section illustrates recent research in Cloud data storage correctness. There are few approaches which make use of soft client applications without use of extra hardware. Kamara at el. [1] propose a template of complete secure storage structure without mentioning much on implementation of components involved. Pearson et al. [8] describe a privacy manager which protects the data being stolen or misused. Though both of these approaches reduce the burden of extra hardware cost from Cloud user/provider, the performance is compromised to some extent, which can be improved with third party auditor (TPA) and/or additional hardware such as cryptographic coprocessors.

Recently, few researchers have proposed approaches based on third party auditor (TPA). Wang at el. [2] propose an approach which enables public auditability for Cloud data storage security through external TPA, without demanding local copy of data or imposing extra online burden on Cloud. Gowrigolla at el. [12] outline a data protection scheme with public auditing which allows data to be stored in encrypted form on Cloud server without loss of accessibility or functionality for authorized users. Homomorphic token are being utilized by Wang at el. [3] and Tribhuvan at el. [10] to achieve data storage correctness. Wei at el. [4] develop an auditing scheme which seeks data storage security, computation and privacy preservation with the help of probabilistic sampling technique and verifier technique. Chuang at el. [5] design an Effective Privacy Protection Scheme (EPPS) which provides privacy protection according to user's demand and also claim to achieve performance.

Temper-proof cryptographic coprocessors configured by trusted third party are proposed by Itani at el. [6] and Ram at. el. [11] to solve the problem of securely processing confidential data in Cloud infrastructure based on various trust levels. Cheng at el. [7] make use of Trusted Platform Module (TPM) with sealed storage ability. While enjoying the benefits of improved performance through extra hardware, these approaches pose cost burden on Cloud users/providers side. Security issues for cross-Cloud environment are addressed by Li at el. [9]. Xu at el. [13] address the security problem in the direction of securing document service. Yu at el. [14] argue that the Cloud data security problem should be solved from data life cycle perspective.

As every proposal discussed here has its own way of understanding the problem of data storage correctness, they do not handle the problems from all facets. For an instance, ignoring dynamic nature of Cloud or adding unnecessary cost on user part may distract the users from Cloud.

3 Data Storage Security Model

In this section, we propose a data storage security model, which intends to solve the data security problem from multiple facets. The first part outlines the design goals which we aim to achieve and the second part describes the proposed model.

3.1 Design Goals

To propose a Data Storage Security Model for Cloud Computing, our design is expected to achieve following security goals: (a) Storage Correctness (b) Different levels of encryption (c) Lightweight: Low computation and communication overhead (d) Incorporating the issue of Cloud dynamism (e) Duplicate copy of original data should not be generated (g) No assumption of file type or file properties. (h) Optionally, keeping track of changes made by other data users. Added to this list, traditional security goals such as availability, reliability, efficient retrieval and data sharing should not be compromised.

3.2 The Proposed Model

There are main three stakeholders of our model. (i) Cloud data owner (CDO), who generates and owns the data. Possessing all rights about file operation, it can pass on the same to other Cloud data users. (ii) Cloud data user (CDU), who uses the data generated by CDO based on the rights issues, CDU can in turn pass the rights available to it to other CDUs. (iii) Cloud service provider (CSP), which is the central core component of the whole system. It also acts as a data warehouse for CDO and CDU. Figure 1 illustrates the proposed model. The directions of arrows show the path of data flow. The operations indicated by the numbers (mentioned on each arrow) are as follows. (1) CDO/CDU stores/updates his (encrypted) data into Cloud. (2) CDO/CDU retrieves/downloads his data (in encrypted form) from Cloud. (3) CDO/CDU verifies data stored on Cloud for integrity. (4) CDO issues a Coupon to CDU (so that the later can download/retrieve data from Cloud and decrypt it). (5) CDO also issues a part of the Coupon to CSP (which can be used by CSP to allow data user's request to use data). The overall functioning of the proposed model is divided among following four phases.

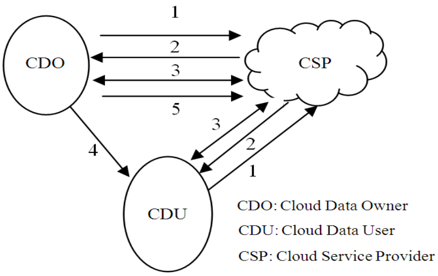

Fig. 1. Proposed Cloud Data Storage Security Model

(A) Registration Phase: CDO and CDU register themselves to Cloud before they start accessing data by providing their unique identification (Customer_ID) and password type code (Customer_Code). This information will be stored in Cloud Customer Registration Master Table (table 1) maintained by the Cloud, for future customer verification by CSP. (B) Pre-Storage Phase: Prior to storing (encrypted) data into the

Table 1. Cloud Customer Registration Master

Field Name	Field Detail
Customer_ID	Unique Identification of each Customer on the Cloud, (Primary key)
Customer_Code	Pass code of the customer to access data on Cloud
Secret_Question	In case of forgetting pass code, this secret question can be asked
Secret_Answer	Answer of the secret question
Is_Active	If user wants to de-register himself, this field will be set to 0.

Table 2. Access Control Policy Master

Field Name	Field Detail
File_ID	Unique identification of every file on Cloud. (Primary key)
Owner_ID	Unique Customer Identification as mentioned in table 1. (Foreign key)
Created_Date_Time	Date and time (time stamp) of file creation.
Encryption_Algo_Type	Type of encryption algorithm. (Contains 0 if data is not sensitive)
Hash/MAC_Code	Hash / MAC code based on encoding algorithm selected by CDO.
Owner_Signature	CDO's signature for later verification
SearchWord	Used in case of searching multiple (encrypted) files by CDU or CDO.

Cloud, CDO needs to decide cryptographic primitives such as encryption algo., encoding algo., signature etc. It will be stored on Access Control Policy Master Table (table 2).

(C) Verification Phase: Any time, CDO/CDU can use this phase to check integrity of data, by issuing QUERY to CSP and CSP returns answer in form of code REPLY, which will be compared by CDO's locally stored code of the same file (or can be re-computed). The integrity of the data is considered to be protected if they are same. (D) Grant Rights Phase: CDO issues a coupon (table 3) to CDU and intimate to CSP by sending few of the coupons' information (table 4) to CSP.

Table 3. Full Coupon Format

Field Name	Field Detail
File_ID	Unique file Identification.
Owner_ID	Unique CDO Identification.
User_ID	Unique CDU Identification.
Access_Rights	Rights (E.g. E, V, Z) given by CDO to CDU.
Encryption_Algo_Type	Type of encryption algorithm. (Contains 0 if data is not sensitive)
Symmetric_Key	Key for encryption. (Contains NULL if data is not sensitive)
Encoding_Algo_Type	Encoding algorithm type. (Contains 0 if encoding is not selected)

In case of granting access right to other CDU, CDO can write a line to table 4. Granting and revoking rights will be performed through simple SQL query such as:
Set Access_Rights to 'V' where File_ID='101034' and User_ID='101';

Table 4. Access Control Policy Detail

Field Name	Field Detail
File_ID	File for which the access right is to be given.
Owner_ID	CDO's unique ID.
User_ID	CDU's unique ID.
Access_Rights	Access right (E.g. E for edit, V for view, Z for revoking grant)

Based on sensitivity, users' data can be divided into three categories (a) *Not sensitive* (fully trusted model) (b) *Highly sensitive* data (not trusted model) and (c) *Moderately sensitive data* (partial cryptographic primitives). In case of some legitimate issue, we optionally provide an audit trail in form of log file (table 5) to CDO which tells him the changes made by other CDUs in the file owned by him.

Table 5. Access Control Log

Field Name	Field Detail
File_ID	Unique file ID.
Customer_ID	Unique user ID which updated the file.
Updation_Date_Time	Date and time (time stamp) of last modification.
FileSize_Before_Update	This is the size of the file before modification.
FileSize_After_Update	This is the size of the file after modification.

As mentioned earlier, confidentiality and integrity are the two of the main goals to be achieved in Cloud computing. Both the operations in our model are achieved as mentioned beneath.

(A) Encryption Process (Performed offline): Performed at CDO's site or CDU's site, they can choose encryption algorithm along with appropriate key or they can use their custom-designed algorithms, too. Two broadly known options for encryption viz. symmetric key encryption (e.g. AES) and asymmetric key encryption (e.g. RSA) may be used here. The keys are to be stored and maintained by the data owner, per file, locally. (Alternatively, we can use a trusted third party, which takes care of storage and maintenance of these keys.) (B) Verifying Data Integrity: Simply downloading the data for integrity verification is not a practical solution due to expensiveness in I/O cost and unsafe files transfer across the network and may lead to new vulnerabilities [16]. Moreover, legal regulations, such as (HIPAA) [17], further demand the outsourced data not to be leaked to external parties (e.g. TPA). So applying encryption before outsourcing is the most preferred way to mitigate the privacy concern.

Along with MD5 and MAC, Proof of storage [15] is widely used protocol for the purpose of checking integrity of data stored on remote server. The algorithms can be run any number of times as user wants, and they do not result into too much communication or computations overhead. It produces a very small amount of information (irrespective of the size of the data file) which can be exchanged between user and Cloud, any number of times.

Next comes, the process of sharing data. As shown in figure 1, step 4 and step 5 illustrate the sharing requirement. In step 4, CDO issues a Coupon (as shown in table 3) to CDU (so that the CDU can download (retrieve) and decrypt the file). In step 5, CDO also issues a part of the Coupon to CSP (which can be used by CSP to allow data user's request to use data) as shown in table 4. Taking care of Cloud dynamism in terms of revoking access right and deregistering a user from Cloud is just a matter of executing a query as shown earlier.

4 Achieving the Proposed Design Goals

After proposing the scheme for data storage correctness, now it's time to cross-verify the goals that we planned in problem statement. (a) Storage Correctness: CDO can anytime request CSP for data correctness. CDO issues a QUERY to CSP and CSP gives REPLY in form of the code Hash/MAC_Code stored in table 2. CDO recomputes the same (off line) and compares it with the code received to check data integrity protection. (b) Encryption based on sensitivity of data: Data is divided among three categories based on its sensitivity viz. (i) not sensitive, (ii) moderately sensitive and (iii) highly sensitive. CDO specifies encryption/encoding algorithm based on data sensitivity in the field Encryption_Algo_Type and Encoding_Algo_Type in table 5. (c) Lightweight: Main two functionalities viz. confidentiality and integrity are to be achieved through encryption and encoding algorithms. We propose these two operations to be performed offline on the premise of CDO or CDU. To check integrity of data i.e. storage correctness, only a small data (hash or MAC code) is to be exchanged among CSP and CDO, which is independent of the file size. (d) Dynamism: Granting and revoking access rights to or from CDU is just a matter of writing a small SQL query and updating table 4, as shown earlier. After every modification file size is updated in table 5 by CDO/CDU. Log table 5, gives information about the trail of changes made by Cloud users. Increase in number of users may not be a matter of great disturbance for Cloud as it merely increases the rows in table 1. (e) No data duplication: Without asking local copy of data, correctness can be measured even data is in encrypted form. The decryption is also done offline at the site of CDO/CDU. (f) Legal and compliance issues: In case of any legal dispute, if data owner is under investigation, enforcement agencies cannot get data directly from CSP. Sometimes there may be a difference of opinion on who made changes to the shared file. Access control log file (table 5) keeps complete track of changes being made by various user on a file. CSP may give this log to CDO or other regulatory bureau upon their quest. (g) Data type or format: We do not make any assumption regarding the data file type or its format. Researches such as [13] heavily rely on data type and format. Apart from this, our solution does not bring in new vulnerabilities. There is no additional online burden on Cloud in terms of data transportation. Dynamic nature of Cloud storage has also been taken care of.

5 Implementation of Core Components

A variety of cryptographic primitives can be used to implement core components and services of our proposed Cloud Storage Security Model. Our model uses various cryptographic services viz. (a) encryption (b) encoding (c) authentication. (a) User may select one of the two types of encryption techniques for the proposed model: first, symmetric key encryption for data encryption by CDO/CDU, or second, asymmetric key encryption for encrypting the coupon to be transferred between CDO and CDU. (b) Encoding in form of hash function is required for verifying the data integrity of the encrypted data by CDO/CDU to make sure that there has not been any unauthorized alteration in the data. (c) Apart from these, we may use other cryptographic primitives such as digital signature which can be used to authenticate the CDO/CDU by CSP.

There are symmetric encryption algorithms, such as AES [18], DES [19], triple DES (TDEA) [30] etc, which can be used for private/single-key encryption. To achieve various encryption depths, we can use AES (or other algorithms) with variable key size. Handling the coupons require utmost care, as they contain sensitive information such as Symmetric_Key and the coupon is transferred from CDO to CDU over the public network. One option is to transfer the entire coupon in encrypted form using public key encryption. The public key of CDU is used to encrypt the coupon and CDU decrypts the same using its private key. RSA [20] or ECC [21] can be few of the better choices for public key encryption. Another type of recently proposed cryptographic technique is attribute-based encryption (ABE [22] [23]). Another alternative symmetric key approach to avoid encrypting the coupon is to run Diffie-Hellman [31] key exchange algorithm between CDO and CDU to share the common Symmetric_Key. In this case, coupon may be optionally encrypted. Proof of storage [24] [15], MD5 [25], Message Authentication Code-MAC or SHA-1 [26] protocol can be used for encoding the encrypted data. For the purpose of Cloud consumer authentication, digital signature, Kerberos [27] or X.509 [28] can also be used.

In our implementation phase, we aim to work out the communication and computation cost of one or more of these cryptographic algorithms in Cloud environment along with their impact on confidentiality, integrity and authentication.

6 Conclusion

In this paper, we proposed a Cloud Data Storage Model. This model aims to achieve lightweight storage correctness along with provision to consider dynamic nature of Cloud. We emphasized that the proposed design prototype is to be only meant for usage as an illustration form. The main part of this model is to develop a client application in open source standard, which is to be downloaded by Cloud customer (CDO/CDU) from CSP in the beginning of entire process (one time only), and provides all the functionalities related various encryption-decryption, key management, encoding, decoding, integrity checking functions such as MAC, Hash, and Proof of storage protocols.

References

1. Kamara, S., Lauter, K.: Cryptographic Cloud Storage. In: Proceedings of the 14th International Conference on Financial Cryptograpy and Data Security, FC 2010, pp. 136–149. Springer, Heidelberg (2010)
2. Wang, C., Wang, Q., Ren, K., Lou, W.: Privacy-preserving public auditing for data storage security in cloud computing. In: 2010 Proceedings IEEE INFOCOM, vol. 54(2), pp. 1–9 (2010)
3. Wang, C., Wang, Q., Ren, K., Lou, W.: Ensuring data storage security in cloud computing. Cryptology ePrint Archive, Report 2009/081 (2009)
4. Wei, L., Zhu, H., Cao, Z., Jia, W., Vasilakos, A.V.: Seccloud: Bridging secure storage and computation in cloud. In: Distributed Computing Systems Workshops (2010)
5. Chuang, I.H., Li, S.H., Huang, K.C., Kuo, Y.H.: An effective privacy protection scheme for cloud computing. In: 2011 13th International Conference on Advanced Communication Technology (ICACT), pp. 260–265 (2011)
6. Itani, W., Kayssi, A., Chehab, A.: Privacy as a service: Privacy-aware data storage and processing in cloud computing architectures. In: Proceedings of the 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, DASC 2009, pp. 711–716. IEEE Computer Society, Washington, DC (2009)
7. Cheng, G., Ohoussou, A.: Sealed storage for trusted cloud computing. In: International Conference on Computer Design and Applications (ICCDA), vol. 5, pp. V5-335 –V5-339 (2010)
8. Pearson, S., Shen, Y., Mowbray, M.: A Privacy Manager for Cloud Computing. In: Jaatun, M.G., Zhao, G., Rong, C. (eds.) CloudCom 2009. LNCS, vol. 5931, pp. 90–106. Springer, Heidelberg (2009)
9. Li, W., Ping, L.: Trust Model to Enhance Security and Interoperability of Cloud Environment. In: Jaatun, M.G., Zhao, G., Rong, C. (eds.) CloudCom 2009. LNCS, vol. 5931, pp. 69–79. Springer, Heidelberg (2009)
10. Tribhuwan, M., Bhuyar, V., Pirzade, S.: Ensuring data storage security in cloud computing through two-way handshake based on token management. In: 2010 International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom), pp. 386–389 (2010)
11. Ram, C., Sreenivaasan, G.: Security as a service (sass): Securing user data by coprocessor and distributing the data. In: Trendz in Information Sciences Computing (TISC), pp. 152–155 (2010)
12. Gowrigolla, B., Sivaji, S., Masillamani, M.: Design and auditing of cloud computing security. In: 2010 5th International Conference on Information and Automation for Sustainability (ICIAFs), pp. 292–297 (2010)
13. Xu, J.-S., Huang, R.-C., Huang, W.-M., Yang, G.: Secure Document Service for Cloud Computing. In: Jaatun, M.G., Zhao, G., Rong, C. (eds.) CloudCom 2009. LNCS, vol. 5931, pp. 541–546. Springer, Heidelberg (2009)
14. Yu, X., Wen, Q.: A view about cloud data security from data life cycle. In: 2010 International Conference on Computational Intelligence and Software Engineering (CiSE), pp. 1–4 (2010)
15. Juels, A., Kaliski Jr., B.S.: Pors: proofs of retrievability for large files. In: Ning, P., di Vimercati, S.D.C., Syverson, P.F. (eds.) ACM Conference on Computer and Communications Security, pp. 584–597. ACM (2007)

16. Shah, M.A., Baker, M., Mogul, J.C., Swaminathan, R.: Auditing to keep online storage services honest. In: Proceedings of the 11th USENIX Workshop on Hot Topics in Operating Systems, pp. 11:1–11:6. USENIX Association, Berkeley (2007)
17. 104th United States Congress: Health Insurance Portability and Accountability Act of 1996 (HIPPA) (1996), `http://aspe.hhs.gov/admnsimp/pl104191.htm`
18. Advanced encryption standard (AES) (FIPS pub. 197) (2001)
19. FIPS 46-3: Data Encryption Standard (DES). (fips pub 46-3) (1999)
20. Rivest, R., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 21, 120–126 (1978)
21. Certicom Research. Standards for efficient cryptography, SEC 1: Elliptic curve cryptography, Version 1.0 (2000), `http://www.secg.org/`
22. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, p. 89 (2006)
23. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: Proceedings of the 14th ACM Conference on Computer and Communication Security, CCS 2007, pp. 195–203. ACM, New York (2007)
24. Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z., Song, D.: Provable data possession at untrusted stores. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, CCS 2007, pp. 598–609. ACM, NY (2007)
25. Rivest, R.: The md5 message-digest algorithm (1992)
26. Institute of standards and technology, N.: FIPS 180-2, Secure Hash Standard, Federal Information Processing Standard (FIPS), Publication 180-2. Tech. rep., Department Of Commerce (2002)
27. Neuman, B.M., Miller, S.P., Neuman, B.C., Schiller, J.I., Saltzer, J.H.: Section e.2.1 kerberos authentication and authorization system. Project Athena Technical Plan (1987)
28. Housley, R., Ford, W., Polk, W., Solo, D.: Internet x.509 public key infrastructure certificate and crl profile (1999)
29. Sanka, S., Hota, C., Rajarajan, M.: Secure Data Access in Cloud Computing. In: 2010 IEEE 4th International Conference on Internet Multimedia Services Architecture and Application(IMSAA), pp. 1–6 (2010)
30. Triple data encryption algorithm. Technical Report Federal Information Processing Standard Publication 46-3, standard ANSI X9.52-1998, NIST (1998)
31. Rescorla, E.: Diffie-Hellman Key Agreement Method. RFC2631 (1999)