Protecting data in the cloud – A Customer Approach

[a]M A Mujeeb, [b]S.Shalini
[a]P G Scholar, Department of CSE, Aurora's Scientific Technological & Research Academy, Telangana, India
[b]Assistant Professor, Department of CSE, Aurora's Scientific Technological & Research Academy, Telangana, India

Abstract

Cloud computing is rapidly emerging due to the provisioning of elastic, flexible, and on-demand storage and computing services for customers. Organizations with a low budget can now utilize high computing and storage services without heavily investing in infrastructure and maintenance. however, the loss of control over data and computation raises many security concerns for organizations, thwarting the wide adaptability of the public cloud. the loss of control over data and the storage platform also motivates cloud customers to maintain the access control over data (individual data and the data shared among a group of users through the public cloud).moreover, the privacy and confidentiality of the data is also recommended to be cared for by the customers .the confidentiality management by a customer ensures that the cloud does not learn any information about the customer data. Cryptography is used as a typical tool to provide confidentiality and privacy services to the data. the data are usually encrypted before storing to the cloud. The access control, key management, encryption, and decryption processes are handled by the customers to ensure data security. However, when the data are to be shared among a group, the cryptographic services need to be flexible enough to handle different users, exercise the access control, and manage the keys in an effective manner to safeguard data confidentiality.
the data handling among a group has certain additional characteristics as opposed to two-party communication or the data handling belonging to a single user. The existing, departing, and newly joining group members can prove to be an insider threat violating data confidentiality and privacy. Insider threats can prove to be more devastating due to the fact that they are generally launched by trusted entities. Due to the fact that people trust insider entities, the research community focuses more on outsider attackers. Nevertheless, multiple security issues can arise due to different users in a group.

I. Introduction

The Sedasc Methodology Is Proposed To Provide The Following Services To The Outsourced Data:- Confidentiality; Secure Data Sharing Among The Group; Secure Data From Unauthorized Access Of Valid Insiders Within The Group; And Forward And Backward Access Control To Counter Insiders And Departing Group Users. In The Case Of Sedasc, The File Is Encrypted With K. K Is Generated At The CS And Is Deleted Right After Utilization. The CS Or The User Cannot Reconstruct K Alone. For Confidentiality, The Data Cannot Be Leaked Unless The Attacker Gains Access To K. K

In Its Entirety Is Not Stored Anywhere, And Neither Does It Travel On The Communication Channel. Therefore, The Access To K Is A Difficult Task. Although An Attacker Gets Hold Of The User Share, I.E., Ki', He/She Will Have To Guess The Other Share Correctly. The Guess Or Random Generation Is To Be Made From A Total Of 2^256 − 1 Possible Shares. The Probability of Generating the Correct Share Is (1/ (2256 − 1)) = 8.636 × 10^−78, Which Is Negligible. Moreover, If The Insider Within The Cloud Gets Access To The File, The Absence Of K Will Be A Barrier To Subvert The Confidentiality Of The Data. For Secure Data Sharing, Sedasc Does Not Utilize The Concept Of Reencryption With Multiple Keys. The Encryption Is Done With A Single Symmetric Key. However, The Authorized Users Are Granted Access On The Basis Of Possession Of The Key Share And The Typical Authentication And Authorization Phenomenon. The ACL Lists The Authorized Users With Their Credentials And Corresponding CS Key Shares. After Authentication, The User Share Of The Key Is Used, Along With The CS Share, To Generate K. As The User Share Is Only Possessed By A Valid User, Only A Valid User Can Lead To Successful Encryption/Decryption Of The Data. The Division And Dispersal Of The Key Also Helps Counter The Insider Malicious Users Within The Group. The ACL Is Separately Maintained For Each Group File. Therefore, A Valid Group User Cannot Access The Group File That Is Not Shared With Him/Her. An Attempt To Access An Unauthorized File Is Also Blocked By The Fact That The User Will Not Have The Key Share For That File. Moreover, The ACL Of The Unauthorized File Will Not Contain Any Record For The Malicious User. Furthermore, The Absence Of The Entire Key With The User And The ACL Collectively Ensures The Forward And Backward Access Control For The Data.Most Of The Data Forwarding Schemes Are Dependent On The El-Gamal Cryptosystem And Bilinear Pairing. The Aforesaid Schemes Require The Reencryption Of The Data Each Time The Access To The Data Is Requested By Any User Other Than The Owner. The El-Gamal Cryptosystem Is Computationally Intensive. Moreover, Reencryption At Each Access Adds To The Overhead. The Sedasc Methodology Utilizes Symmetric Encryption, And The Access To Multiple Users Is Achieved Through Key Management, As Explained In The Preceding Section. Therefore, The Overhead Of The Sedasc Methodology Is Fairly Less As Compared With The Traditional El-Gamal-Based Reencryption Systems.

II. Research work from literature

Xu Et Al. [8] Proposed A Certificate less Proxy Reencryption (CL-PRE) Scheme For Securely Sharing The Data Within A Group In The Public Cloud. In The CL-PRE Scheme, The Data Owner Encrypts The Data With The Symmetric Key. Subsequently, The Symmetric Key Is Encrypted With The Public Key Of The Data Owner. Both The Encrypted Data And The Key Are Uploaded To The Cloud. The Encrypted Key Is Reencrypted By The Cloud That Becomes Decryptable By The User's Private Key. The Public–Private Keys Generated In The Proposed Scheme Are Not Based On The Certificates. The User's Identity Is Used To Generate The Public– Private Key Pair. The Proxy Reencryption Is Based On Bilinear Pairing And The BDH That Makes The CL-PRE Scheme Computationally Intensive. The Computational Cost Of The Bilinear Pairing Is High As Compared With The Standard Operations In Finite Fields. Seo Et Al.

[6] Introduced A Mediated Certificateless Encryption Approach For Data Sharing In The Public Cloud That Avoids Bilinear Pairing. In The Proposed Scheme, The Cloud Generates The Public–Private Key Pairs For All Of The Users And Transmits The Public Keys To All Of The Participating Users. Partial Decryption Is Performed At The Cloud. Due To The Fact That Key Management And Partial Decryption Are Handled By The Cloud, User Revocation Is Easier To Handle. However, The Proposed Scheme Treats The Public Cloud Both As A Trusted And Untrusted Entity At The Same Time. From A Security Perspective, It Is Not Recommended To Shift The Key Generation Process To The Shared Multitenant Public Cloud Environment. Moreover, The Decryption Is Performed Twice In The System That Reduces The Advantage Of Not Pairing To Some Extent. Khan Et Al. [2] Also Utilized The Elgamal Cryptosystem And Bilinear Pairing For The Sharing Of Sensitive Information In The Cloud. The Proposed Scheme Uses A Trusted Third Party As A Proxy That Performs The Compute-Intensive Operations Of Key Generation, Reencryption, And Managing Access To The Data. However, The Computational Complexities Of Bilinear Pairing Still Exist In The System. Chen And Tzeng [9] Proposed A Methodology Based On The Shared Key Derivation Method For Securing Data Sharing Among A Group. The Methodology Uses A Binary Tree For The Computation Of Keys. However, The Computational Cost Of The Proposed Scheme Is High As The Rekeying Mechanism Is Heavily Employed In The Proposed Scheme. Moreover, The Scheme Is Not Tailored For Public Cloud Systems Because Certain Operations Require Centralized Mediations. A Similar Rivest– Shamir–Adleman (RSA)-Based Approach Was Also Proposed In. However, The Scheme Was Vulnerable Against Collusion Attacks.

III. System Architecture

The Sedasc Methodology Works With Three Entities As Follows: -Users; A Cryptographic Server (CS); And The Cloud. The Data Owner Submits The Data, The List Of The Users, And The Parameters Required For Generating An Access Control List (ACL) To The CS. The CS Is A Trusted Third Party And Is Responsible For Key Management, Encryption, Decryption, And Access Control. The CS Generates The Symmetric Key And Encrypts The Data With The Generated Key. Subsequently, For Each User In The Group, The CS Divides The Key Into Two Parts Such That A Single Part Alone Cannot Regenerate The Key. Successively, The Original Key Is Deleted Through Secure Overwriting. One Part Of The Key Is Transmitted To The Corresponding User In The Group, Whereas The Other Part Is Maintained By The CS Within The ACL Related To The Data File. The ACL Is Generated Through The Parameter Submitted By The Data Owner. The Encrypted Data Are Subsequently Uploaded To The Cloud For Storage On Behalf Of The User. The User Who Wishes To Access The Data Sends A Download Request To The CS. The CS, After Authenticating The Requesting User, Receives The Portion Of The Key From The User And Subsequently Downloads The Data File From The Cloud. The Key Is Regenerated By Operating On The User Portion Of The Key, And The Corresponding CS Maintained Portion For That Particular User. The Data Are Decrypted And Sent Back To The User. For A Newly Joining Member, The Two Portions Of The Key Are Generated, And The User Is Added To The ACL. For A

Departing Member, The Record Is Deleted From The ACL. The Departing Member Cannot Decrypt The Data On Its Own As He/She Only Possesses A Portion Of The Key. Similarly, No Frequent Decryption And Reencryption Are Needed In Case Of Changes In The Group Membership. Moreover, Sedasc Can Be Used With The Mobile Cloud Computing Paradigm In Addition To Conventional Cloud Computing Due To The Fact That Compute-Intensive Operations Are Performed By The CS.

IV. Challenges

A Single Key Shared Between All Groups Members Will Result In The Access Of Past Data To A Newly Joining Member. The Aforesaid Situation Violates The Confidentiality And The Principle Of Least Privilege. Likewise, A Departing Member Can Access Future Communication. Therefore, In Group-Shared Data, The Inside Members Might Generate The Issue Of Backward Access Control (A New User Accessing Past Data) And Forward Access Control (A Departing User Accessing Future Data). The Simple Solution Of Rekeying (Generating A New Key, Decrypting All The Data, And Reencrypting With The New Key) Does Not Prove To Be Scalable For Frequent Changes In The Group Membership. A Separate Key For Every User Is A Cumbersome Solution. The Data Must Be Separately Encrypted For Every User In Such A Scenario. The Changes In The Data Require Decryption Of All Of The Copies Of The Users And Encryption Again With The Modified Contents. The Existing And Legitimate Group Members Might Show Illegitimate Behavior To Manipulate The Data .The Presence Of The Entire Symmetric Key With A User Allows A Malicious User To Turn Into An Insider Threat .The Data Can Be Decrypted, Modified, And Reencrypted By A Malicious Insider Within A Group. Consequently, A Legitimate User In The Group May Access Certain Unauthorized Files Within The Group. On The Other Hand, It Is Necessary For A User To Possess A Key To Conduct Various Operations On The Data. The Possession Of The Key Also Implicitly Proves The Legitimacy Of A User To Operate On The Data. Nevertheless, Simultaneously Dealing With Both The Issues Related To The Key Is An Important Issue That Needs To Be Addressed Effectively.

SeDaSC Entities

The SeDaSC methodology has the following entities:-

Cloud: The Cloud Provides Storage Services To The User. The Data On The Cloud Need To Be Secured Against Privacy Breaches. The Confidentiality Of The Data Is Ensured By Storing Encrypted Data Over The Cloud. The Cloud In The Sedasc Methodology Only Involves Basic Cloud Operations Of File Upload And Download. Therefore, No Changes At The Protocol Or Implementation Level On The Cloud Are Required.

CS: The CS Is A Trusted Party And Is Responsible For Security Operations, Such As Key Management, Encryption, Decryption, The Management Of The ACL For Providing Confidentiality, And Secure Data Forwarding Among The Group. The Users Of Sedasc Are Required To Be Registered With The CS To Obtain The Security Services. The CS Is Assumed To Be A Secure Entity In The Proposed Methodology. The CS Can Be

Maintained By An Organization Or Can Be Owned By A Third-Party Provider. However, The CS Maintained By An Organization Will Generate More Trust In The System.

Users: The Users Are The Clients Of The Storage Cloud. For Each Data File, One User Will Be The Owner Of The File, Whereas The Others In The Group Will Be The Data Consumers. The Owner Of The File Decides The Access Rights Of The Other Group Members. The Access Rights Are Granted And Revoked Based On The Decision Of The Owner. The Access Rights Are Managed By The CS In The Form Of An ACL File. A Separate ACL Is Maintained For Each Of The Data Files.

V. Conclusion on Cryptographic Keys

The Sedasc Methodology Maintains A Single Cryptographic Key For Each Of The Data Files. However, After Encryption/Decryption, The Whole Key Is Not Stored And Possessed By Any Of The Involved Parties. The Key Is Partitioned Into Two Constituent Parts And Are Possessed By Different Entities. The Following Are The Keys That Are Used Within Sedasc:-

Symmetric Key K: K Is A Random Secret Generated By The CS For Each Of The Data Files. The Length Of K In Sedasc Is 256 Bits, As Is Recommended By Most Of The Standards Regarding Key Length For Symmetric Key Algorithms (Skas). However, The Length Of The Key Can Be Altered According To The Requirements Of The Underlying SKA. K Is Obtained In A Two-Step Process. In The First Step, A Random Number R Of Length 256 Bits Is Generated Such That R = {0, 1}256. In The Next Step, R Is Passed Through A Hash Function That Could Be Any Hash Function With A 256-Bit Output. In Our Case, We Used Secure Hash Algorithm 256 (SHA-256). The Second Step Completely Randomizes The Initial User-Derived Random Number R. The Output Of The Hash Function Is Termed As K And Is Used In Symmetric Key Encryption [E.G., The Advanced Encryption Standard (AES)] For Securing The Data.
CS Key Share Ki: For Each Of The Users In The Group, The CS Generates Ki, Such That Ki = {0, 1} ^256. Ki Serves As The CS Portion Of The Key And Is Used To Compute K Whenever An Encryption/Decryption Request Is Received By The CS. Moreover, It Is Ensured By Comparison That The Distinct Ki Is Generated For Every File User.
User Key Share Ki': Ki' Is Computed For Each Of The Users In The Group As Follows: Ki' = K XOR Ki. Ki' Serves As The User Portion Of The Key And Is Used To Compute K When Needed.

References

[1]. Abbas And S. U. Khan, "A Review On The State-Of-The-Art Privacy Preserving Approaches In E-Health Clouds," IEEE J. Biomed. Health Informat., Vol. 18, No. 1, Pp. 1431– 1441, Jul. 2014.

[2]. N. Khan, M. M. Kiah, S. A. Madani, M. Ali, And S. Shamshir-Band, "Incremental Proxy Re-Encryption Scheme For Mobile Cloud Computing Environment," J. Supercomput., Vol. 68, No. 2, Pp. 624–651, May 2014.

[3]. 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.

[4]. K. Alhamazani Et Al., "An Overview Of The Commercial Cloud Monitoring Tools: Research Dimensions, Design Issues, Stateof-The-Art," Computing, DOI: 10.1007/S00607-014-0398-5, 2014, To Be Published.

[5]. D. Chen Et Al., "Fast And Scalable Multi-Way Analysis Of Massive Neural Data," IEEE Trans. Comput., DOI: 10.1109/TC.2013.2295806, 2014, To Be Published.

[6]. S. Seo, M. Nabeel, X. Ding, And E. Bertino, "An Efficient Certificateless Encryption For Secure Data Sharing In Public Clouds," IEEE Trans. Knowl. Data Eng., Vol. 26, No. 9, Pp. 2107–2119, Sep. 2013.

[7]. N. Khan, M. L. M. Kiah, S. U. Khan, And S. A. Madani, "Towards Secure Mobile Cloud Computing: A Survey," Future Gen. Comput. Syst., Vol. 29, No. 5, Pp. 1278–1299, Jul. 2013.

[8]. L. Xu, X. Wu, And X. Zhang, "CL-PRE: A Certificateless Proxy Reencryption Scheme For Secure Data Sharing With Public Cloud," In Proc. 7th ACM Symp. Inf. , Comput. Commun. Security, 2012, Pp. 87–88

[9]. Y. Chen And W. Tzeng, "Efficient And Provably-Secure Group Key Management Scheme Using Key Derivation," In Proc. IEEE 11th Int. Conf. Trustcom, 2012, Pp. 295–302.

[10]. S. U. R. Malik, S. K. Srinivasan, S. U. Khan, And L. Wang, "A Methodology For OSPF Routing Protocol Verification," In Proc. 12th Int. Conf. Scalcom, Changzhou, China, Dec. 2012, Pp. 1–5.

[11]. Cloud Security Alliance, "Security Guidelines For Critical Areas Of Focus In Cloud Computing V3.0," 2011.

[12]. L. Moura And N. Bjrner, "Satisfiability Modulo Theories: An Appetizer," In Proc. Formal Methods, Found. Appl., Vol. 5902, Lecture Notes In Computer Science, 2009, Pp. 23–36.

[13]. P. Gutmann, "Secure Deletion Of Data From Magnetic And Solid-State Memory," In Proc.
6th USENIX Security Symp. Focusing Appl. Cryptography, 1996, P. 8.

[14]. T. Murata, “Petri Nets: Properties, Analysis And Applications,” Proc. IEEE, Vol. 77, No. 4, Pp. 541–580, Apr. 1989.

[15]. Y. Chen, J. D. Tygar, Andw. Tzeng, “Secure Group Key Management Using Unidirectional Proxy Re-Encryption Schemes,” In Proc. IEEE INFOCOM, Pp. 1952–1960.