Review of the Various Optimized Access Control Techniques for Big Data in Cloud Environment

Roslin Dayana K.
Assistant Professor
R.M.D Engineering College,
Gummidipoondi Taluk, Thiruvalluvar Dist.

Vigilson Prem M., PhD
Professor
R.M.D Engineering College,
Gummidipoondi Taluk, Thiruvalluvar Dist.

ABSTRACT

Cloud computing is an information technology (IT) domain that enables efficient access to shared and private collection of configurable system resources. It provides higher-level services that can be very quickly provisioned at a greater rate with minimum amount of effort for management, mostly over the Internet. Due to the high complexity and huge volume, outsourcing ciphertexts to a cloud is deemed to be one of the most effective approaches for big data storage and access. Verifying the access legitimacy of a user and securely updating a ciphertext in the cloud based on a new access policy designated by the data owner are two critical challenges. The access policy update is important for enhancing security and dealing with the dynamism caused by user join and leave activities. In this paper, the two different approaches developed recently to provide the secure, verifiable and flexible access control of Big data storage in cloud are discussed to solve the above challenges. The working and drawbacks of different schemes developed in the past for the access control are also discussed.

General Terms

Cloud Computing, Big Data, Cryptosystem

Keywords

NTRU cryptosystem, RSA cryptosystem, Attribute-Based Encryption (ABE), Proxy Re-Encryption (PRE), Access Control List, Role Based Access Control (RBAC)

1. INTRODUCTION

In the past two decades, huge and various type of data called big data are stored in cloud environment for the benefit of different end-users like virtualized desktop users, non-technical end users, cloud choreographers and cloud service providers. These data need to be accessed effectively with optimal job performance and can be quickly accessed for any request that arrives. And, the data that is to be stored on the cloud should have the access protection from the malicious or cheating behaviors of the cloud. This paper discusses about the two recent innovations to improve the access control process in the cloud and also how security can be provided during this access.

2. LITERATURE SURVEY

A survey on various data access control and security methods in cloud is done and the next section in this paper discusses the benefits of two main approaches found recently and the findings of the drawbacks of approaches developed earlier.

3. DATA SECURITY

3.1 A Secure and Verifiable Access Control Scheme for Big Data Storage in Cloud [1]

The first approach to be discussed is a new NTRU decryption algorithm. NTRU stands for Nth Degree Truncated Polynomial Ring Units. NTRU is the first public key cryptosystem not based on factorization or discrete logarithmic problems. The new NTRU decryption algorithm is used to overcome the decryption failures of the original NTRU. This is a secure and verifiable access control scheme based on the NTRU cryptosystem for big data storage in clouds. It allows the cloud server to efficiently update the ciphertext when a new access policy is specified by the data owner. The data owner can validate the update to counter against cheating behaviors of the cloud.

Related Work

Clouds can be classified into two major categories: i) public clouds with each being a multi-tenant environment shared with many other tenants, and ii) private clouds with each being a single-tenant environment dedicated to a single tenant. For example, the IBM cloud was proposed as a public one for the data management of banking.

Secret sharing [3] is a powerful technique to protect the big data in cloud storage. The most related work to the new NTRU decryption scheme are New efficient and practical verifiable multi-secret sharing schemes [5] and A verifiable multi-secret sharing scheme based on cellular automata [4], whose verification procedure can resist potential attacks such as collusion and cheating. In [5], two schemes were proposed, namely Scheme-I and Scheme-II, based on the homogeneous linear recursion and the RSA cryptosystem, in which the homogeneous linear recursion is used to construct the secret share and reconstruct the secret, and RSA is used to verify the users' access legitimacy. The difference between these two schemes is the users in Scheme-I mutually verify each other's legitimacy without seeking help from public values while in Scheme-II the users need the help of public values. In [4], the authors presented a verifiable multi-secret sharing scheme based on cellular automata, which is used to construct the secret share and reconstruct the secret with a linear computational complexity, and the RSA cryptosystem, which is used for verification.

In these schemes, as multiple users mutually verify each other using multiple RSA operations, a very high computational overhead occurs. In addition, the classic asymmetric crypto solutions would be broken by quantum computing; that is, these traditional verification methods cannot satisfy the verification requirements with respect to quantum computing, which is made closer to reality by IBM in 2015 in the paper, Demonstration of a quantum error detection code using a square lattice of four superconducting qubits [6]. Thus, a new verification method is needed to meet the future requirements. For this purpose, the new NTRU cryptosystem is developed to counter the quantum computing attack. Delegation is a popular approach for policy update. In the paper, Plutus: Scalable secure file sharing on untrusted storage [7], a user generates a new private key using its previous private key, and then delegates the new private key to a local authority for access policy update. In the paper Sirius: Securing untrusted storage [8], a procedure called "ciphertext delegation" was designed for the third party to're-encrypt' the ciphertext to a more restrictive policy using only public information. These two approaches cannot satisfy the security requirements because they delegate the private key/ciphertext for a new access policy that is more restrictive than the old one - in authors perspective, the access policy to a ciphertext might need to be relaxed as time goes for many real-world applications.

Working of the New NTRU Cryptosystem

In the new NTRU cryptosystem [1], first, a plaintext data is bound to a secret that is shared by all legitimate users of the data based on (t, n)-threshold secret sharing, and a message certificate is computed for the data based on the NTRU encryption; the ciphertext is produced from both the shared secret and the message certificate. Second, the legitimacy of a user for accessing the data is verified by both the data owner and at least t – 1 other legal users of the data, and the information provided by other users for the plaintext recovery needs to be validated by the user to prevent against cheating behaviors. Third, the plaintext data can be obtained when at least t - 1 other users participate in the recovery process and provide correct information for the data recovery, based on (t, n)-threshold secret sharing.

Last, the access policy of the data and the secret shares bound to the data can be dynamically changed by the data owner, and the update of the ciphertext is conducted by the cloud server without the need of downloading the previous ciphertext from the cloud to the data owner. Meanwhile, the data owner can verify whether the ciphertext stored in the cloud is correctly updated.

Algorithm

The Improved NTRU Decryption

```
Input: cipher text $e$, secret key $\{f, f_p\}$.
Output: plaintext $m$;
The decryptor computes $a = e * f$;
$\Gamma = \max\{|\max_{0\le i\le N-1}\{a_i\}|, |\min_{0\le i\le N-1}\{a_i\}|\}$;
$\tau = \left\lfloor \frac{\Gamma}{q/2} \right\rfloor$;
If $\tau = 0$
    $m = a * f_p \pmod{p}$.
Else
    For $0 \le i \le N-1$,
        Compute $\gamma = \left\lfloor \frac{|ai|}{q/2} \right\rfloor$ ;
        If $\gamma = 0$
            $a_i' = a_i$ and $c_i^{(1)} = c_i^{(2)} = \cdots = c_i^{(\tau)} = 0$;
        Else If $a_i \ge 0$
            $a_i' = a_i - ((q-1)/2)\gamma$;
            $c_i^{(1)} = c_i^{(2)} = \cdots = c_i^{(\gamma)} = (q-1/2)$;
            $c_i^{(\gamma+1)} = a_i'$;
            $c_i^{(\gamma+2)} = \cdots = c_i^{(\tau)} = 0$;
        Else
            $a_i' = a_i + ((q-1)/2)\gamma$;
            $c_i^{(1)} = c_i^{(2)} = \cdots = c_i^{(\gamma)} = -(q-1/2)$;
            $c_i^{(\gamma+1)} = a_i'$;
            $c_i^{(\gamma+2)} = \cdots = c_i^{(\tau)} = 0$;
        EndIf
    EndFor
    $m' = a' * f_p + c^{(1)} * f_p + \cdots + c^{(\tau)} * f_p \pmod{p}$;
EndIf
Output plaintext $m'$.
```

System Model

A cloud storage system is applicable for both public and private clouds as shown in Figure 1.

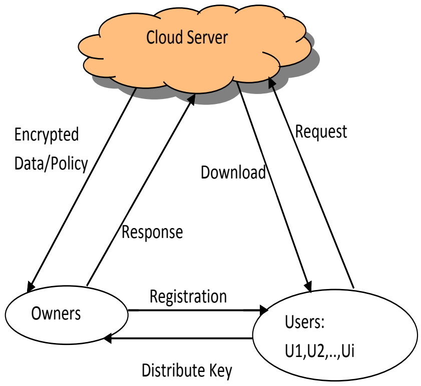

Figure 1: A System Model

It consists of the following three types of entities: Cloud Server, Data Owner (owners), and Data User (users).

Cloud server. A cloud server provides spaces for data owners to store their outsourced ciphertext data that can be retrieved by the users. It is also responsible for updating the ciphertexts when the data owner changes its access policy.

Owners. A data owner designates the access policy for its data, encrypts the data based on the access policy before outsourcing the data to the cloud server, and requests the cloud server to update the encrypted data when a new access policy is adopted. It can also check whether the ciphertext at the cloud server is correctly updated.

Users. Each user is assigned with a sub-key for an encrypted data the user is eligible to access. To decrypt the ciphertext, the user's eligibility must be verified by at least t - 1 other users that are also eligible to access the data. The information provided by the t - 1 verifiers must be validated by the user for correct message decryption based on the (t, n)-threshold secret sharing.

For a piece of data to be stored in a cloud, the data owner generates a public key and privacy key pair, defines an access policy, and computes a sub-key for each potential user based on the policy. Then, the data owner produces a message certificate for the data, and stores the encrypted data with the access policy in the cloud. When a user needs to use the data, it solicits help from other users to recover the data. The cloud server can update the encrypted data with a new policy is designated by the data owner.

Benefits

(i) The NTRU encryption algorithm allows the data owner and eligible users to effectively verify the legitimacy of a user for accessing the data, and

(ii) It allows a user to validate the information provided by other users for correct plaintext recovery. Rigorous analysis indicates that this scheme can prevent eligible users from cheating and resist various attacks such as the collusion attack.

3.2 Flexible Data Access Control Based on Trust and Reputation in Cloud Computing [2]

The second approach to be discussed is a scheme to control data access in cloud computing based on trust evaluated by the data owner and/or reputations generated by many reputation centers in a flexible manner by applying Attribute-Based Encryption (ABE) and Proxy Re-Encryption (PRE). They integrated the concept of context-aware trust and reputation evaluation into a cryptographic system to support various control scenarios and strategies.

Related Work

Many solutions have been proposed for protecting data access in the cloud. Access Control List (ACL) based solutions suffer from the drawback that computation complexity grows linearly with the number of data-groups [7] or the number of users in the ACL [8]. Role Based Access Control (RBAC) cannot flexibly support various data access demands that rely on trust [9]. In recent years, access control schemes based on Attribute-Based Encryption (ABE) were proposed for controlling cloud data access based on attributes to enhance flexibility [10-12, 13, 14, 15, 16]. However, the computation cost of these solutions is generally high due to the complexity of attribute structure. The time spent on data encryption, decryption and key management is more than symmetric key or asymmetric key encryptions.

Most of the existing schemes cannot support controlling cloud data access by either the data owner or access control agents or both. This fact greatly influences the practical deployment of existing schemes. Current research is still at the stage of academic study.

A. Access Control on Encrypted Data

Different cryptographic mechanisms are applied to realize access control on encrypted data. By adopting a traditional symmetric key cryptographic system, the data owner can classify data with similar ACLs into a data-group before outsourcing to CSP, and then encrypts each data-group with a symmetric key. The symmetric key will be distributed to the users in the ACL, so that only the users in the ACL can access the corresponding group of data [7]. The main drawback of this approach is that the number of keys managed by the data owner grows linearly with the number of data-groups. The change of trust relationship between one user and the data owner could make the symmetric key revoked, which impacts other users in the same ACLs and increases the burden of key management. Thus, this solution is impractical in many real application scenarios.

Another approach is based on the combination of traditional symmetric key and public key cryptographic systems [8]. The data owner first specifies an ACL for a data, and then encrypts the data with a symmetric key, which is encrypted with the public keys of users in the ACL. Therefore, only the users in the ACL can recover the data using their private keys. The main drawback of this approach is that the cost for encrypting the symmetric key grows linearly with the number of users in the ACL. This approach cannot efficiently handle frequent changes of trust relationships, either.

RBAC has been applied in cloud computing. It provides flexibility on access control management at a level that corresponds closely to an organization’s policy and structure. Zhou et al. proposed a secure RBAC-based cloud storage system where the access control policies are enforced by Role-Based Encryption (RBE) [9]. This RBE scheme enforces RBAC policies on encrypted data stored in the cloud with an efficient user revocation mechanism, so that only the users with appropriate roles specified by a RBAC policy can decrypt the data. Wang et al. proposed a dynamic role based access control framework by integrating trusted computing with RBAC in cloud computing [17]. We can find many RBAC mechanisms for cloud computing in the literature [33, 34], but most of them cannot flexibly satisfy various data access demands that request trust, especially for the same role. Fine-grained access control inside a role cannot be supported.

B. User Revocation

User revocation is not a trivial task. The key problem is that the revoked users still retain the keys issued earlier, and thus can still decrypt ciphertexts. Therefore, whenever a user is revoked, the re-keying and re-encryption operations need to be executed by the data owner to prevent the revoked user from accessing the future data. For example, when ABE is adopted to encrypt data, the work in [17] proposed to require the data owner to periodically re-encrypt the data, and re-distribute new keys to authorized users. This approach is very inefficient due to the heavy workload introduced to the data owner.

C. Reputation System

Building a mutual trust relationship between users and cloud platform is the key to implement new access control methods in cloud. There are many reputation management systems available nowadays. Some work proposes to compose many services together based on trust and select services based on reputation. Lin et al. proposed a mutual trust based access control (MTBAC) model. It takes both user's behavior trust and cloud service credibility into consideration. Trust relationships between users and CSPs are established by mutual trust mechanism. However, most existing reputation management systems didn’t consider how to control personal data access based on reputation over the cloud. Access control at CSP based on trust and reputation was seldom studied.

Working of the Data Access Control Cryptosystem

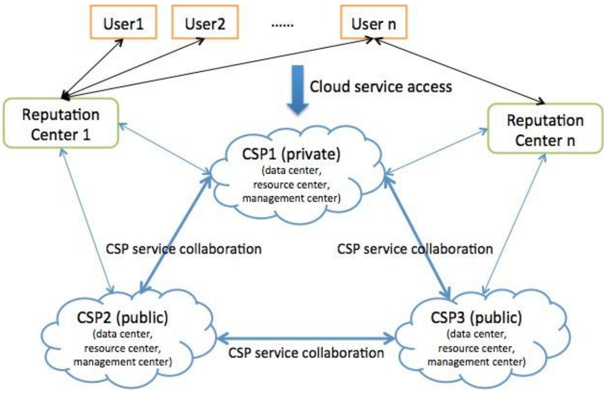
User1
User2
......
User n
Cloud service access
Reputation Center 1
Reputation Center n
CSP1 (private)
(data center, resource center, management center)
CSP service collaboration
CSP service collaboration
CSP2 (public)
(data center, resource center, management center)
CSP service collaboration
CSP3 (public)
(data center, resource center, management center)

Figure 2: A System Model

The system model involves three various kinds of entities, as illustrated in Figure 2: the cloud user that interacts with CSPs for consuming numerous services (e.g., data storage and data access). The user can be a data owner or a data requester; the reputation center (RC) that has functions and capability that the user does not have and is trusted to generate and provide reputation certificates for system entities regarding different data access contexts; the CSP that can be either private for specific users or public for all users and other CSPs. The private and public CSPs can collaborate to provide a service requested by a user. For example, when a private CSP cannot satisfy a user's demand, it could collaborate with other public or private CSPs. The data owned by a system entity and stored at the CSP could be accessed by another system entity during the fulfillment of a cloud service.

Users are not only human beings, but also CSPs. Each CSP has its own data center for data storage, a resource center that can offer numerous services and a management center that is responsible for service request and provision.

The system design holds the following assumptions. RC is a trusted party for reputation generation in different data access contexts. It can collect sufficient information to conduct accurate reputation evaluation, thus provide accurate reputation information of each system entity. Multiple RCs could exist in the system. An insurance company can operate RC. It compensates loss of data disclosure. To earn reputation and business profits, RC should behave honestly (based on an analysis with game theory). The data owner (that is also a cloud user) has a trustworthy personal device that can directly control personal data access based on individual trust evaluation on different system entities, e.g., according to social networking experiences.

The CSP offers data storage services. But it could be curious to seek the privacy of other parties based on stored data and may disclose it. CSP provides stored data to a requester according to the instruction of RC and/or the data owner due to business incentive. RC is always available for registration and authorization of data access rights. But RC is not allowed to access the stored data by CSPs. RCs and CSPs don't collude with each other due to business reasons since collusion may make both lose profits. The communications between the system entities are secured by applying an existing security protocol. Each cloud user registers at its delegating RCs with a unique identifier and personal data access policies. Our scheme follows existing regulations, e.g., relevant identities and qualification certificates (e.g., health physician certifications) should be registered and verified before executing our scheme. We further assume context-aware trust evaluation is applied to support our scheme. We only consider the trust or reputation required in the context of data access. Notably, the trust level sufficient for different data access could be different. In different contexts, different trust evaluation algorithms could be applied for supporting access control. The data owner can choose RCs based on their reputations, which can be evaluated based on data owner feedback, the QoS of RC services, and so on.

Algorithm

Assume that user 1 (u_1) saved its sensitive personal data at CSP, while user 2 (u_2) requests to access it with the authorization of u_1 and one RC.

Step 0: System setups by calling Setup.

Step 1: u_1 generates an encryption key K and separates it into two parts K_0 and K_1. It encrypts data M with the secret key K to get CT. It generates the data access policy AA about individual trust level threshold, public reputation threshold for accessing M. u_1 uploads the encrypted data CT, policy AA and encrypted K_1 (CK_1) by applying Encrypt1 and encrypted K_0 (CK_0) by applying Encrypt0 to CSP; u_1 also sends AA to RC.

Step 2: u_2 would like to access u_1's data by requesting CSP. The CSP checks the validity of its ID and the package of encrypted K to decide if forwarding this request to u_1 and/or RC if it is not in the greylist. Based on the content in AA, the CSP decides whether to contact u_1 and/or RC.

Step 3: If RC is contacted, RC evaluates u_2's reputation and checks if it satisfies with M's access policy AA. Based on the reputation level, RC generates $rk\_RC \rightarrow u_2$ if access is allowed; meanwhile, if u_1 is contacted, it checks the eligibility of u_2 in order to generate a personalized secret key sk-(TL, u_1, u_2) for u_2 to decrypt CK_0.

Step 4: RC issues $rk\_RC \rightarrow u_2$ to the CSP that re-encrypts the CK_1 to get (pk_{u2},K_1) if the re-encryption was never conducted; meanwhile, u_1 issues $sk\_(TL, u_1, u_2)$ to u_2.

Step 5: CSP allows u_2 to access requested data by providing corresponding encrypted data CT and encrypted keys (CK_1 and CK_0) to u_2.

Step 6: u_2 decrypts CK_1 and CK_0 with the issued secret keys from u_1 and its private key sk_{u2}. By combining K_1 and K_0, u_2 can get the complete K to decrypt CT and get M.

Step 7: u_1 re-evaluates the trust based on past and newly accumulated experiences regarding the data access context. If u_2 has been issued the secret keys and is not eligible at present, u_1 will put them into its underlying data access greylist and inform the CSP. RC can re-generate reputation of different entities based on newly collected data. If RC indicates that u_2 doesn't satisfy with access policy AA, RC will inform the CSP to block u_2's access to u_1's data.

The greylist is data-oriented since different data access may request different trust levels. Its content is dynamically upgraded based on timely trust and reputation.

4. CONCLUSION

In this paper, we reviewed a collection of schemes to control cloud data access based on trust and reputation. The recent scheme namely a secure and verifiable access control scheme incorporates with a trust/reputation management framework for securing cloud computing by applying ABE, PRE-and a reputation-based revocation mechanism. In future, this scheme can be applied in real applications of cloud data protection in eHealth services and Internet of Things. We also reviewed an improved NTRU cryptosystem to overcome the decryption failures of the original NTRU and a secure and verifiable access control scheme based on the improved NTRU to protect the outsourced big data stored in a cloud. NTRU scheme allows the data owner to dynamically update the data access policy and the cloud server to successfully update the corresponding outsourced ciphertext to enable efficient access control over the big data in the cloud. It also provides a verification process for a user to validate its legitimacy of accessing the data to both the data owner. Designing a more secure, privacy preserving, and practical scheme for big data storage in a cloud is an extremely challenging problem is the future enhancement of this technique.

5. REFERENCES

[1] Chunqiang Hu, Wei Li, Xiuzhen Cheng, Jiguo Yu, Shenling Wang, Rongfang Bie. “A Secure and Verifiable Access Control Scheme for Big Data Storage in Cloud ”, IEEE Transactions on Big Data, Vol pp, issue 99, Feb 2017.

[2] Zheng Yan, Xueyun Li, Mingjun Wang, Athanasios V. Vasilakos “Flexible Data Access Control Based on Trust and Reputation in Cloud Computing”, IEEE Transactions on Cloud Computing, Vol 5, issue 3, July-Sept. 1 2017.

[3] A. Shamir, “How to share a secret”, Communications of the ACM, Vol 22, No. 11, pp. 612-613, 1979.

[4] Z.Eslami and J.Z.Ahmadabadi, “A verifiable multi-secret sharing scheme based on cellular automata”, Information Sciences, Vol 180, No. 15, pp. 2889-2894, 2010. Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[5] M. H. Dehkordi and S. Mashhadi, “New efficient and practical verifiable multi-secret sharing schemes,” Information Sciences, vol. 178, no. 9, pp.2262–2274, 2008.

[6] A. C´orcoles, E. Magesan, S. J. Srinivasan, A. W. Cross, M. Steffen, J. M. Gambetta, and J. M. Chow, “Demonstration of a quantum error detection code using a square lattice of four superconducting qubits”, Nature communications, vol. 6, pp. 1–10, 2015.

[7] M. Kallahalla, et al., “Plutus: Scalable secure file sharing on untrusted storage”, Proc. of the USENIX Conference on File and Storage Technologies (FAST), 2003, pp. 29–42.

[8] E. Goh, H. Shacham, N. Modadugu, D. Boneh, “Sirius: Securing untrusted storage”, Proc. of NDSS, 2003, pp. 131–145.

[9] L. Zhou, V. Varadharajan, M. Hitchens, “Achieving secure role-based access control on encrypted data in cloud storage”, IEEE Trans. on Information Forensics and Security, vol. 8, no. 12, pp. 1947-1960, 2013.

[10] S. Yu, C. Wang, K. Ren, W. Lou, “Achieving secure, scalable, and fine-grained data access control in cloud computing”, Proc. of the IEEE INFOCOM, 2010, pp. 534–542.

[11] G. Wang, Q. Liu, J. Wu, M. Guo, “Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers”, Computers & Security, vol. 30, no. 5, pp. 320–331, 2011.

[12] S. Yu, C. Wang, K. Ren, W. Lou, “Attribute based data sharing with attribute revocation”, Proc. of the ACM ASIACCS, 2010, pp.261–270.

[13] G. Wang, Q. Liu, J. Wu, “Hierarchical attribute-based encryption fine-grained access control in cloud storage services”, Proc. of the 17th ACM CCS, 2010, pp. 735–737.

[14] M. Zhou, Y. Mu, W. Susilo, J. Yan, “Piracy-preserved access control for cloud computing”, Proc. of IEEE TrustCom11, 2011, pp.83-90.

[15] Z. Wan, J. Liu, R.H. Deng, “HASBE: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing”, IEEE Trans. on Info. Forensics & Security, vol. 7, no.2,pp.743-754,2012.

[16] Z. Yan, Trust Management in Mobile Environments – Usable and Autonomic Models, IGI Global, Hershey, Pennsylvania,2013.

[17] W. Wang, J. Han, M. Song, X. Wang, “The design of a trust and role based access control model in cloud computing”, in Proc. of 6th International Conference on Pervasive Computing and Applications,2011,pp.300-334.

[18] M. Pirretti, P. Traynor, P. McDaniel, B. Waters, “Secure attribute based systems”, Journal of Computer Security, vol.18,no.5,pp.799–837,2010.