Volume 5, Issue 4, 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

An Anonymous Network to Access the Cloud by Anonymous Routing Protocol with Multiple Routes for Communications in Heterogeneous Networks

Vikram Neerugatti*, K Kusuma, K Geethavani

Department of CSE & SVCET,
India

Abstract— Using ***cloud storage; users can remotely store their data and enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in cloud computing a formidable task, especially for users with constrained computing resources. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. To access the cloud we need to mind the security issues like, the Active and passive attacks of the Networks, etc. Thus, enabling Anonymous network for cloud is critical importance, so that users can access data in an efficient manner and be worry free. To securely introduce an effective Anonymous protocol, the accessing process of Cloud should bring in no new vulnerabilities toward user data privacy, and introduce no additional online burden to user. In this paper, we propose an Anonymous Routing Protocol with Multiple Routes (ARMR) which creates the anonymous network for Cloud to communicate in the Heterogeneous Networks. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient.***

Keywords— Data Access, cloud computing, Anonymous Cloud, cloud computing network, cloud Security.

I. INTRODUCTION

CLOUD computing has been envisioned as the next generation information technology (IT) architecture for enterprises, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk [2]. As a disruptive technology with profound implications, cloud computing is transforming the very nature of how businesses use information technology. One fundamental aspect of this paradigm shifting is that data are being centralized or outsourced to the cloud. From users' perspective, including both individuals and IT enterprises, storing data remotely to the cloud in a flexible on-demand manner brings appealing benefits: relief of the burden for storage management, universal data access with location independence, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc., [3].

Now a day's cloud computing is a rationally developed technology to store data from more than one client. Cloud computing is an environment that enables users to remotely store their data. Remote backup system is the advanced concept which reduces the cost for implementing more memory in an organization. It helps enterprises and government agencies reduce their financial overhead of data management.

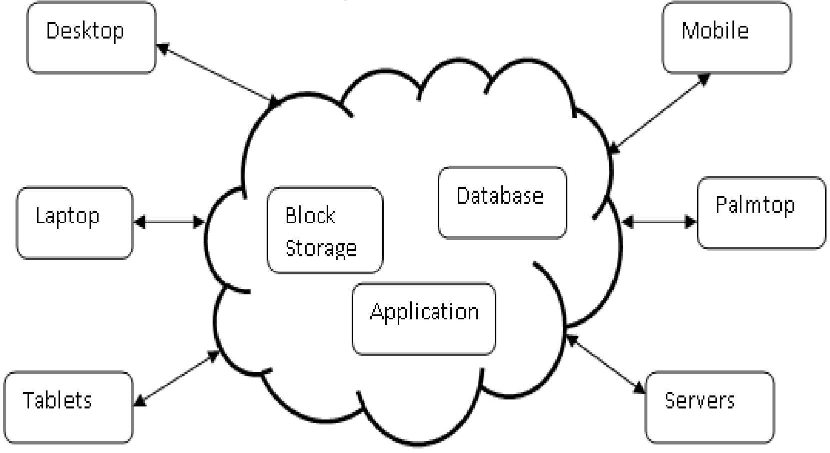

Fig1: Example diagram for data sharing with cloud storage.

They can archive their data backups remotely to third party cloud storage providers rather than maintain data centres on their own. An individual or an organization may not require purchasing the needed storage devices. Instead they can store their data backups to the cloud and archive their data to avoid any information loss in case of hardware / software failures. Even cloud storage is more flexible, how the security and privacy are available for the outsourced data becomes a serious concern.There are three objectives to be main issue **Confidentiality** – preserving authorized restrictions on information access and disclosure. The main threat accomplished when storing the data with the cloud. **Integrity** – guarding against improper information modification or destruction.

Availability – ensuring timely and reliable access to and use of information.

While cloud computing makes these advantages more appealing than ever, it also brings new and challenging security threats toward users' outsourced data. Since cloud service providers (CSP) are separate administrative entities, data outsourcing is actually relinquishing user's ultimate control over the fate of their data. As a result, the correctness of the data in the cloud is being put at risk due to the following reasons. First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity [4]. Examples of outages and security breaches of noteworthy cloud services appear from time to time [5], [6], [7]. Second, there do exist various motivations for CSP to behave unfaithfully toward the cloud users regarding their outsourced data status. For examples, CSP might reclaim storage for monetary reasons by discarding data that have not been or are rarely accessed, or even hide data loss incidents to maintain a reputation [8], [9], [10]. In short, although outsourcing data to the cloud is economically attractive for long-term large-scale storage, it does not immediately offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede the success of cloud architecture. As users no longer physically possess the storage of their data, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted [11]. In particular, simply downloading all the data for its integrity verification is not a practical solution due to the expensiveness in I/O and transmission cost across the network. Besides, it is often insufficient to detect the data corruption only when accessing the data, as it does not give users correctness assurance for those un accessed data and might be too late to recover the data loss or damage. Considering the large size of the outsourced data and the user's constrained resource capability, the tasks of auditing the data correctness in a cloud environment can be formidable and expensive for the cloud users [12], [8]. Moreover, the overhead of using cloud storage should be minimized as much as possible, such that a user does not need to perform too many operations to use the data (in additional to retrieving the data). In particular, users may not want to go through the complexity in verifying the data integrity. Besides, there may be more than one user accesses the same cloud storage, say in an enterprise setting. For easier management, it is desirable that cloud only entertains verification request from a single designated party. To address these problems like the security attacks, our work utilizes the technique of ARMA protocol, which enables users to communicate in secure manner compared to the existing protocols.

The rest of the paper is organized as follows: Section II presents the related works. Section III lays out our problem statement. Then, we provide the detailed description of our proposed work in Section 3. Section 4, finally gives the concluding remark of the whole paper.

II. RELATED WORK

Ateniese et al. [9] are the first to consider public audit ability in their "provable data possession" (PDP) model for ensuring possession of data files on entrusted storages. They utilize the RSA-based homomorphism linear authenticators for auditing outsourced data and suggest randomly sampling a few blocks of the file. However, among their two proposed schemes, the one with public audit ability exposes the linear combination of sampled blocks to external auditor. When used directly, their protocol is not provably privacy preserving, and thus may leak user data information to the external auditor. Jules et al. [11] describe a "proof of irretrievability" (PoR) model, where spot-checking and error-correcting codes are used to ensure both "possession" and "retrievability" of data files on remote archive service systems. However, the number of audit challenges a user can perform is fixed a priori, and public audit ability is not supported in their main scheme. Although they describe a straightforward Merkle-tree construction for public PoRs, this approach only works with encrypted data. Later, Bowers et al. [18] propose an improved framework for POR protocols that generalizes Juels' work. Dodis et al. [29] also give a study on different variants of PoR with private auditability. Shacham and Waters [13] design an improved PoR scheme built from BLS signatures [19] with proofs of security in the security model defined in [11]. Similar to the construction in [9], they use publicly verifiable homomorphic linear authenticators that are built from provably secure BLS signatures. Based on 372 IEEE TRANSACTIONS ON COMPUTERS, VOL. 62, NO. 2, FEBRUARY 2013 Fig. 3. Comparison on auditing time between batch and individual auditing, when _-fraction of 256 responses are invalid: Per task auditing time denotes the total auditing time divided by the number of tasks. The elegant BLS construction, a compact and public verifiable scheme is obtained. Again, their approach is not privacy preserving due to the same reason as [9]. Shah et al. [15], [10] propose introducing a TPA to keep online storage honest by first encrypting the data then sending a number of precomputed symmetric-keyed hashes over the encrypted data to the auditor. The auditor verifies the integrity of the data file and the server's possession of a previously committed decryption key. This scheme only works for encrypted files, requires the auditor to maintain state, and suffers from bounded usage, which potentially brings in online burden to users when the keyed hashes are used up. Dynamic data have also attracted attentions in the recent literature on efficiently providing the integrity guarantee of remotely stored data. Ateniese et al. [21] is the first to propose a partially dynamic version of the prior PDP scheme, using only symmetric key cryptography but with a bounded number of audits. In [22], Wang et al. consider a similar support for partially dynamic data storage in a distributed scenario with additional feature of data error localization. In a subsequent work, Wang et al. [8] propose to combine BLS-based HLA with MHT to support fully data dynamics. Concurrently, Erway et al. [23] develop a skip list based scheme to also enable provable data possession with full dynamics support. However, the verification in both protocols requires the linear combination of sampled blocks as an input, like the designs in [9], [13], and thus does not support privacy-preserving auditing. In other related work, Sebe et al. [30] thoroughly study a set of requirements which ought to be satisfied for a remote data possession checking protocol to be of practical use. Their proposed protocol supports unlimited times of file integrity verifications and allows preset tradeoff between the protocol running time and the local storage burden at the user. Schwarz and Miller [31] propose the first study of checking the integrity of the remotely stored data across multiple distributed servers. Their approach is based on erasure-correcting code and efficient algebraic signatures, which also have the similar aggregation property as the homomorphism authenticator utilized in our approach. Curtmola et al. [32] aim to ensure data possession of multiple replicas across the distributed storage system. They extend the PDP scheme in [9] to cover multiple replicas without encoding each replica separately, providing guarantee that multiple copies of data are actually maintained. In [33], Bowers et al. utilize a two-layer erasure-correcting code structure on the remotely archived data and extend their POR model [18] to distributed scenario with high-data availability assurance. While all the above schemes provide methods for efficient auditing and provable assurance on the correctness of remotely stored data, almost none of them necessarily meet all the requirements for privacy-preserving public auditing of storage. Moreover, none of these schemes consider batch auditing, while our scheme can greatly reduce the computation cost on the TPA when coping with a large number of audit delegations. Portions of the work presented in this paper have previously appeared as an extended abstract in [1]. We have revised the paper a lot and improved many technical details as compared to [1]. The primary improvements are as follows: First, we provide a new privacy-preserving public auditing protocol with enhanced security strength. All the experiments in our performance evaluation for the newly designed protocol are completely redone. Finally, we provide formal analysis of privacy-preserving guarantee and storage correctness, while only heuristic arguments are sketched in [1].

III. PROPOSED WORK

This section presents our ARMR protocol which provides a complete outsourcing solution of data—not only the data itself, but also its integrity checking. We follow a similar definition of previously proposed schemes in the context of remote data integrity checking [9], [11], [13] and adapt the framework for our privacy preserving public auditing system. A public auditing scheme consists of four algorithms (KeyGen, SigGen, GenProof, VerifyProof). KeyGen is a key generation algorithm that is run by the user to setup the scheme. SigGen is used by the user to generate verification metadata, which may consist of digital signatures. GenProof is run by the cloud server to generate a proof of data storage correctness, while VerifyProof is run by the TPA to audit the proof. Running a public auditing system consists of two phases, Setup and Audit:

Setup: The user initializes the public and secret parameters of the system by executing KeyGen, and preprocesses the data file F by using SigGen to generate the verification metadata. The user then stores the data file F and the verification metadata at the cloud server, and delete its local copy. As part of preprocessing, the user may alter the data file F by expanding it or including additional metadata to be stored at server.

. **Audit**: The TPA issues an audit message or challenge to the cloud server to make sure that the cloud server has retained the data file F properly at the time of the audit. The cloud server will derive a response message by executing Gen Proof using F and its verification metadata as inputs. The TPA then verifies the response via Verify Proof. Our framework assumes that the TPA is stateless, i.e., TPA does not need to maintain and update state between audits, which is a desirable property especially in the public auditing system [13]. Note that it is easy to extend the framework above to capture a state full auditing system, essentially by splitting the verification metadata into two parts which are stored by the TPA and the cloud server, respectively. Our design does not assume any additional property on the data file. If the user wants to have more error resilience, he can first redundantly encodes the data file and then uses our system with the data that has error correcting codes integrated.

Privacy-Preserving Public Auditing Scheme

Overview: To achieve privacy-preserving public auditing, we propose to uniquely integrate the Unpadded RSA based homomorphism linear authenticator with random masking technique. In our protocol, the linear combination of sampled blocks in the server's response is masked with randomness generated by the server. With random masking, the TPA no longer has all the necessary information to build up a correct group of linear equations and therefore cannot derive the user's data content, no matter how many linear combinations of the same set of file blocks can be collected. On the other hand, the correctness validation of the block-authenticator pairs can still be carried out in a new way which will be shown shortly, even with the presence of the randomness. Our design makes use of a Unpadded RSA-based HLA, to equip the auditing protocol with public audit ability. Specifically, we use the HLA proposed in [13], which is based on the short signature scheme proposed by Boneh, Lynn, and Shacham (hereinafter referred as BLS signature) [19].

Scheme details: Homomorphic encryption is a form of encryption which allows specific types of computations to be carried out on ciphertext and generate an encrypted result which, when decrypted, matches the result of operations performed on the plaintext.

This is a desirable feature in modern communication system architectures. Homomorphic encryption would allow the chaining together of different services without exposing the data to each of those services, for example a chain of different services from different companies could 1) calculate the tax 2) the currency exchange rate 3) shipping, on a transaction without exposing the unencrypted data to each of those services. Homomorphic encryption schemes are malleable by design. The homomorphic property of various cryptosystems can be used to create secure voting systems, collision-resistant hash functions, private information retrieval schemes and enable widespread use of cloud computing by ensuring the confidentiality of processed data. There are several efficient, partially homomorphic cryptosystems, and a number of fully homomorphic, but less efficient cryptosystems. Although a cryptosystem which is unintentionally homomorphic can be subject to attacks on this basis, if treated carefully homomorphism can also be used to perform computations securely.

In the following examples, the notation $\mathcal{E}(x)$ is used to denote the encryption of the message *x*.

If the RSA public key is modulus m and exponent e, then the encryption of a message x is given by $\mathcal{E}(x) = x^e \bmod m$. The homomorphic property is then

$$\mathcal{E}(x_1)\cdot\mathcal{E}(x_2) = x_1^e x_2^e \bmod m = (x_1 x_2)^e \bmod m = \mathcal{E}(x_1 \cdot x_2)$$

IV. CONCLUSIONS

In this paper, we propose a privacy-preserving public auditing system with anonymous network for data storage security in cloud computing. We utilize the unpadded RSA based homomorphic linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend our privacy-preserving public auditing protocol into a multiuser setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Extensive analysis shows that our schemes are provably secure and highly efficient. We leave the full-fledged implementation of the mechanism on commercial public cloud as an important future extension, which is expected to robustly cope with very large scale data and thus encourage users to adopt cloud storage services more confidently.

REFERENCES

[1] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Storage Security in Cloud Computing," Proc. IEEE INFOCOM '10, Mar. 2010.

[2] P. Mell and T. Grance, "Draft NIST Working Definition of Cloud Computing," http://csrc.nist.gov/groups/SNS/cloudcomputing/ index.html, June 2009.S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," *IEEE Electron Device Lett.*, vol. 20, pp. 569–571, Nov. 1999.

[3] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," Technical Report UCB-EECS-2009-28, Univ. of California,
Berkeley, Feb. 2009.

[4] Cloud Security Alliance, "Top Threats to Cloud Computing," http://www.cloudsecurityalliance.org, 2010. (2002) The IEEE website. [Online]. Available: http://www.ieee.org/

[5] J. Kincaid, "MediaMax/TheLinkup Closes Its Doors," http:// www.techcrunch.com/2008/07/10/mediamaxthelinkup-closesits- doors/, July 2008.*FLEXChip Signal Processor (MC68175/D),* Motorola, 1996.

[6] Amazon.com, "Amazon s3 Availability Event: July 20, 2008," http://status.aws.amazon.com/s3-20080720.html, July 2008.

[7] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 5, pp. 847-859, May 2011.

[8] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 598-609, 2007.

[9] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.

BIOGRAPHIES

Vikram Neerugatti- Received the Master Degree in Computer Science and Engineering from JNTUA of india in 2014 and the M.S. degree in Brainwells University of London, UK. in 2010. Worked as Cantract faculty in NIT Goa. Had 5 years of Teaching experience. Currently working as Assistant Professor at SVCET, Chittoor of India. My research interests are cloud computing and Computer Networks. Had 3 National Conferences and 7 National and International Journals. Attended 4 Workshops and Organized 2 Workshops.

K.Kusuma - Received the Master Degree in Software Engineering from JNTUA of India in 2013. Currently working as Assistant Professor at SVCET, Chittoor of India. My research interests are cloud computing and Bigdata.

K. Geethavani - Received the Master Degree in Software Engineering from JNTUA of India in 2013. Currently working as Assistant Professor at SVCET, Chittoor of India. My research interests are cloud computing and Bigdata. Had 4 years of Teaching Experience. Had 4 Conferences.