Achieve Confidentiality & Authentication for Cloud Data using Hybrid Approach

Nikita Mehta[1] Prof. Khushbu Shah[2]
[1]Research Scholar [2]Assistant Professor
[1,2]Mahakal Institute of Technology Ujjain, M.P, India

Abstract— We are proposing to allow the users to audit the cloud storage with very lightweight and ease of communication and computation cost. The auditing which we are going through our research paper result not only ensures strong cloud storage accuracy guarantee, but it also simultaneously achieves fast data error localization, i.e., the identification of hacker information. And securely we are introducing an effective TPA, which is a auditing process which results in bringing in no new vulnerabilities towards user data privacy, and can introduce no additional online burden to user. In this paper, we are going to propose a secure cloud storage system which supports privacy-preserving public auditing. We will further extend our result to enable the TPA to perform audits for various users certainly and efficiently. This shows the proposed scheme is highly efficient and data modification attack, and useful in even server colluding attacks.

Key words: Aggregation, Bursting, Encryption, Storage, Privacy, Security, Confidentiality, Integrity, TPA

I. INTRODUCTION

In the past few decade we have seen the rise of cloud computing ,in simple form we can define it as an arrangement in which businesses and individual users make use of the hardware, storage, and software of the third party companies which is commonly called cloud providers instead of running their own computing infrastructure .Cloud computing provides the users a platform of having infinite computing resources such as a cloud printer and so on, of which they can use as much or as very less as they need, without having the concern about themselves with precise knowledge how those resources are provided or maintained to the users.

Cloud computing provides a range of services that may vary according to the degree or the requirement to which they conceptual away the details of the core hardware and software from users. At the lowest level of perception, often known to as infrastructure as a service ie IAAS, the provider only virtualizes the hardware and storage while leaving users sensible for conserving the entire software mass from operating system to applications.

A. Scalability:

Cloud computing[1] provides an organizations, both big and small, the opportunity to scale their computing resources whenever they want to satisfy their needs. This is done by either increasing or decreasing the required resources, meaning you're not paying for the resources which you are not utilizing.

B. Availability:

High-availability is, ultimately, the major feature of the cloud. It provides the idea of anywhere and anytime access to services, tools and documents and is the develops [3]of visions of a future with companies with no physical offices or of global industries with completely integrated and unified IT systems.

C. Reliability:

As the implementation of cloud computing carry on to rise, and customers request 24/7 access to their services and data, reliability leftovers a challenge for cloud service providers everywhere[5]. This means it's dangerous for organizations to recognize how best to plan and supply reliable cloud services. If we accept the statistic failures will occur, then the conclusions organizations may want to consider in relative to their cloud services fall into four main classifications:

1) Maximize service availability to customers
2) Minimize the impact of any failure on customers.
3) Maximize service performance.
4) Maximize business continuity.

II. RELATED WORK

Privacy-Preserving Public Auditing for Secure Cloud Storage 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[9] 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[7].

N. Saravanan et. al. [2012] presents An Implementation of RSA Algorithm in Google Cloud using Cloud SQL. Cloud storage concern the user does not have control over data until he has been gain access. To provide control over data in the cloud data-centric security is needed. Before accessing the data it should satisfy the policy rules already defined. So cloud should enforce this scheme by using cryptographic approaches. We utilize RSA algorithm and Google App Engine to provide efficient and secured data storage scheme.

Shobha Rajak et. al. [2012] proposed a model for the integrity check over the cloud computing and we utilize the TPA and digital signature to achieve the integrity concept, in such a way to help the user to verify and examine the data from unauthorized people that manipulate with the cloud or extract from the data. Moreover, we are able to evaluate our work using a windows azure project that involves digital signature coding. As results, we found that our model worked well according to our claims. In future it can be enhancing in the server side updating and data modification. In our paper we decided to concern about the client data storing service in the cloud.[12].

Padmapriya et al.[2013] presents[11] In Cloud computing technology there are a set of important policy issues, which include issues of privacy, security, anonymity,

telecommunications capacity, government surveillance, reliability among others. But the most important between them is security and how cloud provider assures it. This paper analyses the importance of security to cloud. We compared three algorithms namely Data Encryption Standard (DES), RSA, Homomorphic encryption for data security in cloud. They are compared based on four characters; key used scalability, security applied to, and authentication type. In future we are going to propose a backup plan to solve security issues in both cloud providers and cloud consumers[14].

Faraz Fatemi Moghaddam et. al. [2013] presents hybrid asymmetric-key encryption algorithm has been suggested based on RSA Small-e and Efficient RSA according to the security issues in cloud computing environments. In the proposed algorithm, the number of exponents has been increased to three and a dual encryption process has been applied to raise the security level of the algorithm in comparison of original RSA. According the simulation results, the total execution time in HE-RSA was increased up to approximately 50 percent less than the original RSA and this increase may be reasonable and acceptable according to the security level and the efficiency of HE-RSA.

III. PROPOSED WORK

Our security analysis focuses on the adversary model as defined. We also evaluate the efficiency of our scheme via implementation of both file distribution preparation and verification token pre computation. In our scheme, servers are required to operate on specified rows in each correctness, verification for the calculation of requested token. We will show that this "sampling" strategy on selected rows instead of all can greatly reduce the computational overhead on the server, while maintaining the detection of the data corruption with high probability. Suppose nc servers are misbehaving due to the possible compromise or Byzantine failure.

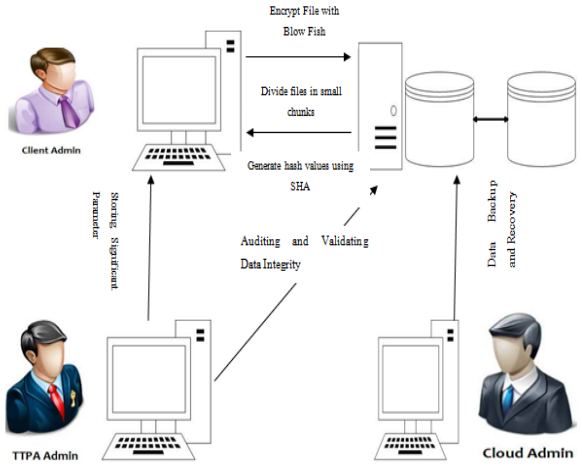

Fig 1 System Architecture

A. *Uploading Steps:*

1) Each user logs on to the workstation using an own ID and Password.
2) No of user connected to a storage array via network.
3) The client computer sends a request to the storage array for storing a file.
4) This file is encrypted by two times.
   – At the time of transferring RSA works which will be encrypt our data.
   – And the second one is MD5 that will be work in data storage array.
5) MD5 need because, threats at storage level include tampering with data, which violates data integrity, and media theft, which compromises data availability and confidentiality.

B. *Downloading Steps:*

1) When the client sends a request form a server, it sends a request, consist of valid ID and Password.
2) The storage array checks the permission and ensures that the user is authorized to use that service.
3) If user is authorized then reply the client machine and give respond.
4) The client computer sends the desired file name that want to access.
5) The storage array decrypts the file and the server automatically allows the client to access the appropriate resources.

IV. BLOWFISH ALGORITHM

Blowfish is a 64-bit symmetric[10] block cipher with inconstant length key. The algorithm operates on the two main parts a key extension part and a data encryption part. The role of key expansion part is to converts a key of at most 448 bits into several sub key arrays totaling 4168 bytes. It is suggestively speedier than most encryption algorithms when they are comparatively implemented on 32-bit microprocessors with sizeable data caches.

While it is impossible to take all Blowfish Blowfish is a 64-bit symmetric block cipher with variable length key. The algorithm operates with two parts: a key expansion part and a data encryption part. The role of key expansion part is to converts a key of at most 448 bits into several sub key arrays totaling 4168 bytes. It is considerably faster than most encryption algorithms when implemented on 32-bit microprocessors with large data caches.

The environment of encryption algorithms is that, once any significant amount of security analysis is completed, it is very undesirable to change the algorithm for performance reasons, thereby invalidating the results of the analysis. Thus, it is imperative to consider both security and performance together during the design phase.

A. *Subkeys:*

Blowfish uses a big number of subkeys. These keys must be precomputed before any data encryption or decryption. The P-array consists of 18 32-bit subkeys: P1, P2,..., P18. There are four 32-bit S-boxes with 256 entries each:
S1,0, S1,1,..., S1,255;
S2,0, S2,1,..,, S2,255;
S3,0, S3,1,..., S3,255;
S4,0, S4,1,..,, S4,255.

B. *Pseudo Code for Blow-Fish Algorithm:*

1) Step1: start the item size.

2) Step 2: 16 rounds are there in blow fish.
3) Step 3: x be the input of 64 bit data element.
4) Step 4: x will be divided into two halves x1 and x2.
5) Step 5: then, for i = 1 to 16:
   X1 = x1 XOR Pi x2 = F(x1) XOR x2
6) Step 6: Swap x2and x2
7) Step 7: After the sixteenth round, swap x1 and x2 again to undo the last swap. Then, x2= x2 XOR P17 and x1 = x1 XOR P18.
8) Step 8: Recombine x1 and x2 to the cipher text
9) Step 9: Decryption in reverse order except p1,p2,…..p18.
10) Step 10: stop

V. CONCLUSION

Traditional symmetric and asymmetric encryption schemes can be leveraged to provide Alice with a secure means through which she can send her message. However, with symmetric schemes each recipient will be in a position to decrypt all cipher-texts that have been encrypted with the same key: Access is too coarse-grained. With asymmetric schemes the encrypting entity needs to explicitly state for whom decryption is permissible: Access is too fine-grained. To reference the different styles of communication, symmetric schemes represent broadcast communication and asymmetric schemes unicast communication. A multicast encryption scheme is required that allows for a more expressive _ne-grained means through which Alice can specify access over data.

The above Blowfish algorithm with AES in positions of the throughput, processing time. More the throughput, more the speed of the algorithm & less will be the power consumption. Finally we can conclude that Blowfish is the best of all. In future work we can perform Hardware Implementation to compare different parameters. This method is used for faster work. Implementations of Blow fish that require the fastest speeds should unroll the loop and ensure that all sub keys are stored in cache.

REFERENCES

[1] Peter Mell and Timothy Grance. NIST special publication 800-145: The NIST de_nition of cloud computing. http://csrc.nist.gov/publications/ nistpubs/800-145/SP800-145.pdf, September 2011.
[2] Michael Armbrust, Armando Fox, Rean Gri_th, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, Dept. of Electrical Engineering and Computer Sciences, University of California at Berkeley, February 2009.
[3] Amazon Web Services LLC. Amazon elastic compute cloud (Amazon EC2). http://aws.amazon.com/ec2/. Retrieved April 23, 2012.
[4] Ian Hickson. Web storage - W3C candidate recommendation. http://www.w3. org/TR/webstorage/, December 2011.
[5] Microsoft. Windows azure virtual machine role. http://www.windowsazure. com/en-us/home/features/virtual-machines/. Retrieved April 23, 2012. 91] Rackspace US, Inc. Rackspace. http://www.rackspace.com/. Retrieved April 23, 2012.
[6] Twitter, Inc. Twitter. http://www.twitter.com/. Retrieved April 23, 2012.
[7] Andrzej M. Goscinski Rajkumar Buyya, James Broberg. Cloud Computing: Prin- ciples and Paradigms. Wiley, 2011.
[8] Jose M. Alcaraz Calero, Nigel Edwards, Johannes Kirschnick, Lawrence Wilcock, and Mike Wray. Toward a multi-tenancy authorization system for cloud services. IEEE Security and Privacy, 8:48{55, 2010.
[9] Google, Inc. Google Apps. http://www.google.com/apps/index1.html. Retrieved April 23, 2012.
[10] N. Saravanan, A. Mahendiran, N. Venkata Subramanian and N. Sairam "An Implementation of RSA Algorithm in Google Cloud using Cloud SQL" Research Journal of Applied Sciences, Engineering and Technology 4(19): 3574-3579, 2012
[11] Kevin Hamlen, Marut Kantarcioglu, latifur Khan and Bhavani Thuraisingham "Security issues for Cloud Computing" Technical Report UTDCS-02-10 February 2010.
[12] Shobha Rajak, Ashok Verma " Secure Data Storage in cloud using Digital Signature Mechanism" International Journal of Advanced Research in Computer Engineering & Technology Volume 1, Issue 4, June 2012
[13] Deyan Chen, Hong Zhao "Data Security and Privacy Protection Issues in Cloud Computing" 2012 International Conference on Computer Science and Electronics Engineering.
[14] Padmapriya et al., International Journal of Advanced Research in Computer Science and Software Engg 3 (4), March - 2013, pp. 255-259.
[15] Parsi Kalpana ,et al, International Journal of Research in Computer and Communication technology, IJRCCT, ISSN 2278-5841, Vol 1, Issue 4, September 2012.
[16] Faraz Fatemi Moghaddam, Maen T. Alrashdan, and Omidreza Karimi "A Hybrid Encryption Algorithm Based on RSA Small-e and Efficient-RSA for Cloud Computing Environments" Journal of Advances in Computer Network, Vol. 1, No. 3, September 2013.
[17] Amandeep Kaur et. al "An Efficient data storage security algorithmusing RSA Algorithm" International Journal of Application or Innovation in Engineering & Management (IJAIEM) Volume 2, Issue 3, March 2013.
[18] Dhaval Patel, M.B.Chaudhari "Data security in cloud computing using digital signature" International Journal For Technological Research In Engineering Volume 1, Issue 10, June-2014
[19] Kevin Fu. Group sharing and random access in cryptographic storage file systems. Technical report, Masters thesis, MIT, 1999.
[20] Dominik Grolimund, Luzius Meisser, Stefan Schmid, and Roger Wattenhofer. Cryptree: A folder tree structure for cryptographic _le systems. In Proceedings of the 25th IEEE Symposium on Reliable Distributed Systems, pages 189{198, Washington, DC, USA, 2006. IEEE Computer Society.

[21] Erik Riedel, Mahesh Kallahalla, and Ram Swaminathan. A framework for evaluating storage system security. In Proceedings of the 1st USENIX Conference on File and Storage Technologies, FAST '02, Berkeley, CA, USA, 2002. USENIX Association.
[22] Paul Stanton, William Yurcik, and Larry Brumbaugh. Protecting multimedia data in storage: A survey of techniques emphasizing encryption. In IS and T/SPIE International Symposium Electronic Imaging / Storage and Retrieval Methods and Applications for Multimedia, pages 18{29, 2005.
[23] Kevin Fu, Seny Kamara, and Tadayoshi Kohno. Key regression: Enabling efficient key distribution for secure distributed storage. In NDSS, 2006.
[24] Michael Backes, Christian Cachin, and Alina Oprea. Secure Key-Updating for Lazy Revocation. In Research Report RZ 3627, IBM Research, pages 327{346. Springer, 2005.
[25] Marina Blanton. Key Management in Hierarchical Access Control Systems, 2007. PhD Thesis, Purdue University, Aug. 2007.
[26] Marina Blanton, Nelly Fazio, and Keith B. Frikken. Dynamic and Efficient Key Management for Access Hierarchies. In Proceedings of the ACM Conference on Computer and Communications Security, 2005.
[27] Saman Zarandioon, Danfeng Yao, and Vinod Ganapathy. K2C: Cryptographic Cloud Storage With Lazy Revocation and Anonymous Access. Technical report, Rutgers University. DCS-tr-688.

