Data Storage using Homomorphic Encryption

Praveen Kumar

Student of Master of Technology (Computer Science), Department of Computer Science & Engineering
Sanskar College of Engineering (SGIT), Ghaziabad

Abstract: ***Organizations are showing great interest in storing data on public clouds. This could be a result of the unprecedented growth of data recorded in the last few years. However the security issues associated with data storage over cloud is a major discouraging factor for potential adopters. Hence the focus of today is to discover cryptographic techniques that will offer additional than privacy. Homomorphic encryption is one such method that has interesting applications in cloud. The objective is to manage and protect the data from the users of a client organization which wants to store the data on untrusted public clouds. In this thesis a hybrid cloud framework is proposed that addresses the privacy and trust issues and provides encrypted storage with public clouds. The proposed method uses Homomorphic Encryption for protecting the user data and uses a modified file updation technique to reduce bandwidth consumption during transfer of large encrypted files.***

Keywords: Cloud Computing, Cryptography, Storage Security in cloud, Homomorphic, Encryption.

1. Introduction

Cloud computing offers a cost-effective solution to manage the IT infrastructure in a flexible and scalable manner. Cloud computing enables software applications, deployment platforms, even the computing resources to be made available on-demand using a pay-as-you-go model. This has drawn a lot of attention towards the domain in recent years. Today a good number of organizations use the cloud for their day to day operations and the adoption rate by others are also high. Hosted applications over the Internet have evolved greatly. The web which originally just consisted of static web pages, today serves as platform for many web applications that ranges from simple note taking tools to computation intensive scientific simulation services. One thing that makes such an approach special that users can outsource data and computation to a remote server that has enough resources to perform the task within much less time than traditionally running an equivalent application on the user's machine. This is also one of the major factors that are driving the research in the cloud computing technologies.

In recent times there have been reports of many security breaches of cloud services such as Dropbox [2,3], Last.fm [4,5], and iCloud [6,7]. A study [8] suggests 72% of the IT professionals blame employees for most data breaches, whereas the rest blame the hackers. It also reveals that 32% data was lost while 18% data was stolen by employees. This increases the concern of insider attack on public clouds.

2. Objective

The objective is to develop a framework using which an organization can store its data on the cloud in a secure manner. The requirements for the framework are as follows. It should be easy to use and should not depend upon security measures taken by the end users e.g. the employees of the organization or users of a service provided by the organization. It should handle all the cryptographic operations within the trusted infrastructure of the organization and then send the encrypted data to the cloud. The public clouds in which the encrypted data is stored should not have the ability to decrypt the content. It should handle file uploads in an efficient manner to reduce bandwidth consumption.

3. Cloud Computing: A Security Perspective

In simple words, Cloud computing can be described as a method that allow resources to be made available over a network in general and Internet in specific. A more formal definition by NIST defines cloud computing as follows. Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [9]. NIST definition also includes five essential characteristics, three service models, and four deployment models for cloud. An overview of the same is presented in Figure 2.1.

4. Cloud Computing Service Delivery Models

Software as a Service (SaaS): It includes services where the consumers are given access to applications deployed on the cloud infrastructure of the providers. The applications are accessed using various device platforms through web browser, or some native applications interface. In this case the consumers do not have the control over the cloud infrastructure. Still they can specify certain limited configuration settings.

Platform as a Service (PaaS): It provides the consumer the ability to deploy an application onto the cloud infrastructure of the provider using development environment supported by the provider. Here the consumers do not have the control over the cloud infrastructure of the provider but do have good control over the applications deployed by them.

Infrastructure as a Service (Iaas) : It provides the consumers ability to provision basic computing resources such as servers, storage, and networks on top of the cloud infrastructure of the provider. In this case the consumers have more control over the resources provisioned and can configure it as needed e.g. the consumers have choice in selecting operating system or installing any software on a provisioned virtual machine. But the consumers still do not manage or control the cloud infrastructure of the provider.

5. Homomorphic Encryption

Homomorphic encryption is a form of encryption which allows specific types of computations to be carried out on ciphertext and obtain an encrypted result which when decrypted gives the result of operations performed on the plaintext[18]. For example, one could add two encrypted numbers and then another could decrypt the result, without either of them being able to find the value of the individual numbers.

Methods that would allow operation on data without knowing the actual content can help in lot of areas. Homomorphic encryption is one such method. Today most systems operate with help of a trusted party. Users have to trust an entity, human or machine to maintain secrecy of their data. But an attack on the trusted party or vulnerability with the system can expose the users secret. Hence the necessity of systems where even the service providers have no detailed knowledge of the users data is growing.

6. Implementation and Results

A prototype to evaluate the working of the proposed framework has been developed in Python. The Paillier homomorphic cryptosystem and associated cryptographic operations are implemented using Sage [32]. The framework has been tested only in local environment as described in section 8.1. This offers details on the performance of the proposed framework with respect to the performance parameter i.e. the file size. For a full-scale cloud implementation a python server process running in the background is needed to keep the framework alive and to provide the client interface.

7. Performance Analysis

In order to understand the performance the different approaches, a plot of change in file sizes for the encrypted and various patch files over the eight instances is presented in figure 8.1. For a given instance the average value of the eight file sizes are considered for each type. It has been observed that the proposed

8. Implementation and Results

Table 5.1: Various file size information for 3 instances.

Files	Encrypted File Size in KB	Patch File Size in KB		
		xdelta	bsdiff	Our Approach
File 1	1032	566	414	362
	2918	1650	1238	138
	5222	2970	2238	666
File 2	570	306	218	150
	1378	770	570	90
	2486	1402	1050	246
File 3	996	546	398	338
	2658	1502	1136	106
	4718	2862	2022	506
File 4	1280	732	556	516
	3800	2176	1652	48
	6600	3784	2872	616
File 5	848	488	368	308
	2260	1296	980	132
	3812	2184	1656	396
File 6	448	260	196	100
	848	484	368	36
	1412	812	612	144
File 7	1172	672	508	480
	3172	1816	1376	256
	5596	3204	2432	616
File 8	2348	1344	1016	928
	7112	4076	3096	540
	10640	6100	4640	932

References

[1] Ponemon research study infographic: Whos minding your cloud? http://www.ca. com/us/collateral/white-papers/na/ponemon-research-study-infographic-whos-minding-your-cloud.aspx, 2013.

[2] Dropbox. https://www.dropbox.com/.

[3] Dropbox confirms it was hacked, offers users help. http://news.cnet.com/8301-1009_3-57483998-83/dropbox-confirms-it-was-hacked-offers-users-help/.

[4] Last.fm. http://www.last.fm/.

[5] Last.fm password security update. http://www.last.fm/passwordsecurity, 2012.

[6] icloud. https://www.icloud.com/.

[7] Another apple disaster: The icloud gets hacked. http://www.forbes.com/sites/ timworstall/2012/08/07/another-apple-disaster-the-icloud-gets-hacked/.

[8] Securing the clouds [infographic]. http://www.tappin.com/blog/2012/12/cloud-security-infographic/, 2012.

[9] Peter Mell and Tim Grance. The NIST definition of cloud computing. Technical report, July 2009.

[10] Deyan Chen and Hong Zhao. Data security and privacy protection issues in cloud computing. In Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on, volume 1, pages 647–651, 2012.

[11] M.A. AlZain, E. Pardede, B. Soh, and J.A. Thom. Cloud computing security: From single to multi-clouds. In System Science (HICSS), 2012 45th Hawaii International Conference on, pages 5490–5499, 2012.

[12] Steven Y. Ko, Kyungho Jeon, and Rams´es Morales. The hybrex model for confidentiality and privacy in cloud computing. In Proceedings of the 3rd USENIX conference on Hot topics in cloud computing, HotCloud'11, pages 8–8, Berkeley, CA, USA, 2011. USENIX Association.

[13] Witold Litwin, Sushil Jajodia, and Thomas Schwarz. Privacy of data outsourced to a cloud for selected readers through client-side encryption. In Proceedings of the 10th annual ACM workshop on Privacy in the electronic society, WPES '11, pages 171–176, New York, NY, USA, 2011. ACM.

[14] A. Patrascu, D. Maimut, and E. Simion. New directions in cloud computing. a security perspective. In Communications (COMM), 2012 9th International Conference on, pages 289–292, 2012.

[15] Wayne Jansen and Timothy Grance. Sp 800-144. guidelines on security and privacy in public cloud computing. Technical report, Gaithersburg, MD, United States, 2011.

[16] https://brilliant.org/wiki/homomorphic-encryption/

[17] S. Subashini and V. Kavitha. Review: A survey on security issues in service delivery models of cloud computing. J. Netw. Comput. Appl., 34(1):1–11, January 2011.

[18] R. Rivest, L. Adleman, and M. Dertouzos. On data banks and privacy homomorphisms. pages 169–177. Academic Press, 1978.

[19] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM, 21(2):120–126, February 1978.

[20] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In Proceedings of CRYPTO 84 on Advances in cryptology, pages 10–18, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

[21] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Proceedings of the 17th international conference on Theory and application of cryptographic techniques, EUROCRYPT'99, pages 223–238, Berlin, Heidelberg, 1999. Springer-Verlag.

[22] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-dnf formulas on ciphertexts. In Proceedings of the Second international conference on Theory of Cryptography, TCC'05, pages 325–341, Berlin, Heidelberg, 2005. Springer-Verlag.

[23] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Proceedings of the 41st annual ACM symposium on Theory of computing, STOC '09, pages 169–178, New York, NY, USA, 2009. ACM.

[24] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In Proceedings of the 31st annual conference on Advances in cryptology, CRYPTO'11, pages 505–524, Berlin, Heidelberg, 2011. Springer-Verlag.

[25] Jean-S´ebastien Coron, David Naccache, and Mehdi Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. In Proceedings of the 31st Annual international conference on Theory and Applications of Cryptographic Techniques, EUROCRYPT'12, pages 446–464, Berlin, Heidelberg, 2012. Springer-Verlag.

[26] Craig Gentry, Shai Halevi, and Nigel P. Smart. Fully homomorphic encryption with polylog overhead. In Proceedings of the 31st Annual international conference on Theory and Applications of Cryptographic Techniques, EUROCRYPT'12, pages 465–482, Berlin, Heidelberg, 2012. Springer-Verlag.

[27] Thorsten Kleinjung, Kazumaro Aoki, Jens Franke, Arjen K. Lenstra, Emmanuel Thom´e, Joppe W. Bos, Pierrick Gaudry, Alexander Kruppa, Peter L. Montgomery, Dag Arne Osvik, Herman Te Riele, Andrey Timofeev, and Paul Zimmermann. Factorization of a 768-bit rsa modulus. In Proceedings of the 30th annual conference on Advances in cryptology, CRYPTO'10, pages 333–350, Berlin, Heidelberg, 2010. Springer-Verlag.

[28] R. Riggio and S. Sicari. Secure aggregation in hybrid mesh/sensor networks. In Ultra Modern Telecommunications Workshops, 2009. ICUMT '09. International Conference on, pages 1–6, 2009.

[29] Raluca Ada Popa, Catherine M. S. Redfield, Nickolai Zeldovich, and Hari Balakrishnan. Cryptdb: protecting confidentiality with encrypted query processing. In Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles, SOSP '11, pages 85–100, New York, NY, USA, 2011. ACM.

[30] Colin Percival. Naive differences of executable code, 2003.

[31] Joshua P. MacDonald. File system support for delta compression. Technical report, 2000.

[32] Sage: Open source mathematics software. http://www.sagemath.org/.