PROVIDING SECURITY BY USING BASTION ALGORITHM FOR COLLEGE MANAGEMENT SYSTEM

[1] J.Jayalakshmi, [2]A.Vamsi Krishna, [3] Dr.B.Sunil Kumar

[1] Sr.Assistant Professor, [2]Assistant Professors, [3] Professor
Dept. of CSE,
Narayana Engineering College, Nellore, A.P, India

Abstract- ABSTRACT: Recent news reveal a powerful attacker which breaks data confidentiality by acquiring cryptographic keys, by means of coercion or backdoors in cryptographic software. Once the encryption key is exposed, the only viable measure to preserve data confidentiality is to limit the attacker's access to the cipher text. This may be achieved, for example, by spreading cipher text blocks across servers in multiple administrative domains thus assuming that the adversary cannot compromise all of them. Nevertheless, if data is encrypted with existing schemes, an adversary equipped with the encryption key, can still compromise a single server and decrypt the cipher text blocks stored therein. In this project, we implement Bastion, a novel and efficient scheme that guarantees data confidentiality even if the encryption key is leaked and the adversary has access to almost all cipher text blocks. We study data confidentiality against an adversary which knows the encryption key and has access to a large fraction of the cipher text blocks. Bastion achieves data confidentiality by combining the use of standard encryption functions with an efficient linear transform. We examine the security of Bastion and we also evaluate its performance.

Keywords- *vitality-based ranking, Reliable*

I.INTRODUCTION

Cloud Computing is Internet ("Cloud") based development and use of computer technology ("Computing"). It is a style of computing in which dynamically scalable and often "virtualized" resources are provided as a service over the internet. Cloud Computing is an emerging technology where data and services reside in massively scalable data centers in the cloud and can be accessed from any connected devices over the internet. Users need not have knowledge of, expertise in, or control over the technology infrastructure "in the cloud" that supports them. It provides limitless "virtualized" resources to users as schemes over the whole Internet, while hiding implementation and platform details.The goal of cloud computing is to apply traditional supercomputing, or high-performance computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer games.

CHARACTERISTICS:

The salient characteristics of cloud computing based on the definitions provided by the National Institute of Standards and Terminology (NIST) are outlined below:On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider. Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).

Resource pooling: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location-independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.

Rapid elasticity: Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.

Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be managed, controlled, and reported providing transparency for both the provider and consumer of the utilized service**.**

Achieve economies of scale – increase volume output or productivity with fewer people. Your cost per unit, project or product plummets. Reduce spending on technology infrastructure. Maintain easy access to your information with minimal upfront spending. Pay as you go (weekly, quarterly or yearly), based on demand. Globalize your workforce on the cheap. People worldwide can access the cloud, provided they have an Internet connection.

Less personnel training is needed. It takes fewer people to do more work on a cloud, with a minimal learning curve on hardware and software issues.Minimize licensing new software. Stretch and grow without the need to buy expensive software licenses or programs.
Improve flexibility. You can change direction without serious "people" or "financial" issues at stake.
Now-a-days cloud service provider offers both massively parallel computing resources and highly available storage at comparatively low costs. As cloud computing becomes universal, a vast amount of data is shared by the users with specified privileges and it is being stored in the cloud, which interpret the ingress rights of the stored data. One analytical dispute of cloud storage services is to secure the data present in cloud.In cloud, security is often better than traditional systems. Security typically improves due to centralization of data, increased security- focused resources. Due to centralize data centre it is possible to improve the level of data security. However, complexity of security is increased when decentralization of data over the wide area of network and various devices are used to get services. Perpetrators were not hindered by the various security measures deployed within the targeted services. For instance, although these services relied on encryption mechanisms to guarantee data confidentiality, the necessary keying material was acquired by means of backdoors or coercion.

If the encryption key is exposed, the only viable means to guarantee data confidentiality is to limit the adversary's access to the cipher text. For example, by spreading the data across multiple administrative domains, in the hope that adversary cannot compromise all of them. However, even if the data is encrypted and dispersed across multiple administrative domains, an adversary equipped with the appropriate keying material can compromise a server in one domain and decrypt cipher text blocks stored therein.In this, data confidentiality is studied against an adversary who knows the encryption key and has access to a large fraction of the ciphertext blocks. The adversary can acquire the key by exploiting flaws or backdoors in the key-generation software, or by compromising the devices that store the keys (e.g., at the user-side or in the cloud). Adversary invalidates the security of most cryptographic solutions, including those that protect encryption keys by means of secret-sharing.

To counter such an adversary, we implement Bastion, a novel and efficient scheme which ensures that plaintext data cannot be recovered as long as the adversary has access to at most all but two ciphertext blocks, even when the encryption key is exposed. Bastion achieves this by combining the use of standard encryption functions with an efficient linear transform. Bastion shares similarities with the notion of all-or-nothing transform.
An AONT is not an encryption by itself, but can be used as pre-processing step before encrypting the data with a block cipher. This encryption paradigm- called AON encryption – was mainly intended to slow down brute-force attacks on the encryption key. Existing AON encryption schemes require at least two rounds of block cipher encryptions on the data: one pre-processing round to create the AONT, followed by another round for the actual encryption. These rounds are sequential, and cannot be parallelized. This results in considerable – often unacceptable – overhead to encrypt and decrypt large files.

Bastion first encrypts the data with one round of block cipher encryption, and then applies an efficient linear post-processing to the cipher text. Bastion requires only one round of encryption – which makes it well suited to be integrated in existing dispersed storage systems. Bastion provides data security when compared to other systems. So the adversary cannot get all the data that is present in cloud. The performance of Bastion is also increased in comparison to a number of existing encryption techniques. Bastion incurs a negligible performance deterioration (less than 5%) when compares to symmetric encryption schemes and considerable improves the performance or existing encryption schemes. It also incurs a negligible overhead when compared to existing semantically secure encryption modes, for e.g., CTR encryption mode.

II. PROBLEM DEFINITION

The motivation behind the project is to provide security for transferring files from a faculty to students.
The main aim of the project is to implement BASTION algorithm for College Management System to provide more security.
The main objective of the project is to generate a secret key for the students in order to download or view a file which are uploaded by the faculty. Admin will provide the secret key to the students for downloading the file. Therefore a file can be viewed by any person who has a secret key which will be provided by the admin.

III.SYSTEM ANALYSIS

In the existing system, if the encryption key is exposed, the adversary can acquire the key by backdoors in the key-generation software, or by compromising devices that store the keys. So the adversary can easily get all the data is being shared among users. There was no data confidentiality. So the end user may also loss originality of data.
DISADVANTAGES:

There was no proper encryption method in the existing system.
There was no proper security in the existing system because adversary can get all the data without loss of originality.
In the proposed system, we implement a novel and efficient scheme called Bastion, which ensures that plaintext cannot be recovered as long as adversary has access to a large fraction of cipher text blocks, even when the encryption key is exposed. To guarantee data confidentiality, the only way is by spreading data across multiple administrative domains, so that the adversary cannot compromise all of them. Bastion first encrypts the data with one round of block cipher encryption, and then applies an efficient linear post-processing to the cipher text. We use polynomial key generation to encrypt the data that is uploaded in Bastion provides more security and privacy when compared to other systems.Bastion improves performance of the system.It provides fast accessibility of data.Bastion also provides deterministic data with zero probability of error occurrences.In this system, to prevent an authorized access, a safety proof of ownership is used to provide the proof that the user is authenticated.Here, both owner and user are authenticated by activating the owner and user by the administrator. By using signature key and private key, owner and user can authenticate. In this system, owner will upload the file in to cloud. Only the authenticated owner can upload the file in to cloud. After uploading the file, owner has to generate the polynomial keys for encryption. By Polynomial key generation, owner can encrypt the data that he uploaded. Finally by using this algorithm, we show that the data that is present in cloud is secured and the performance of the system is also improved.

IV. SYSTEM MODEL

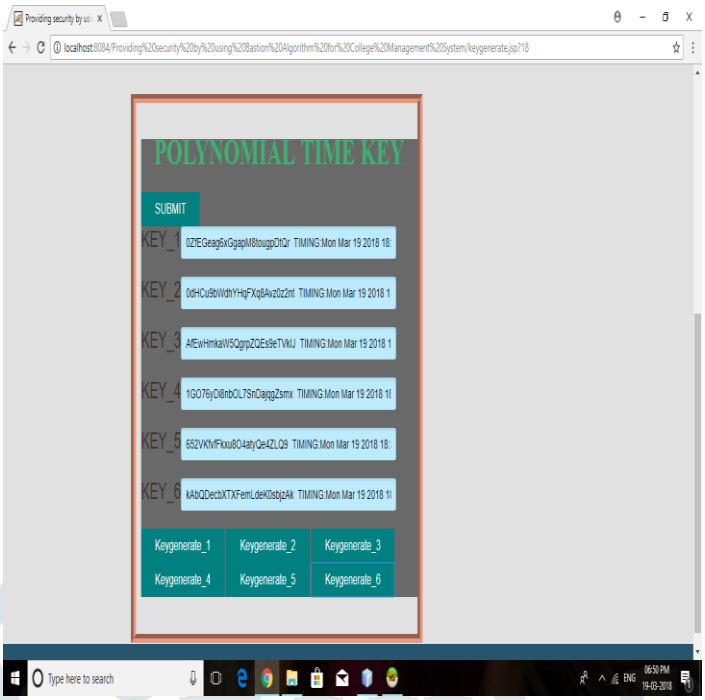
A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. As shown in the figure, the faculty will upload the file or data to the cloud. The file will be encrypted by performing the encryption process on it by using polynomial key generation. Therefore the encrypted file will be uploaded to the cloud. The student will send a request to the college administrator for a file. Then the administrator will send a key to student. Therefore by using that key students can select the required file and the encrypted file will be decrypted. The decrypted file can be downloaded by the student.

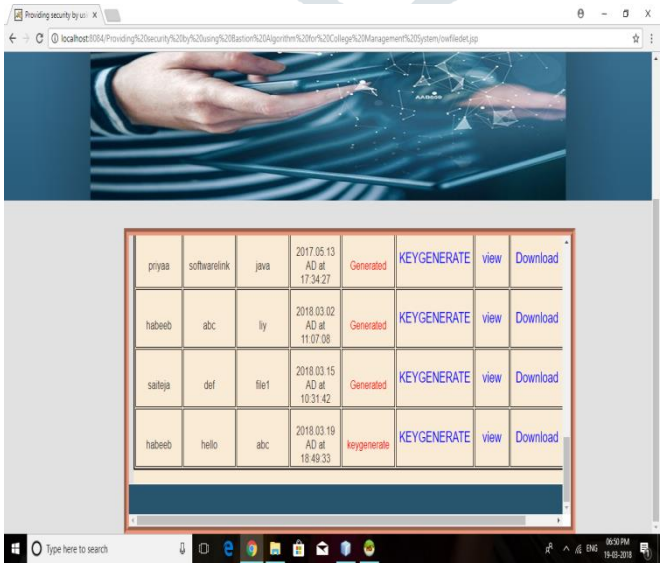
V. IMPLEMENTATION

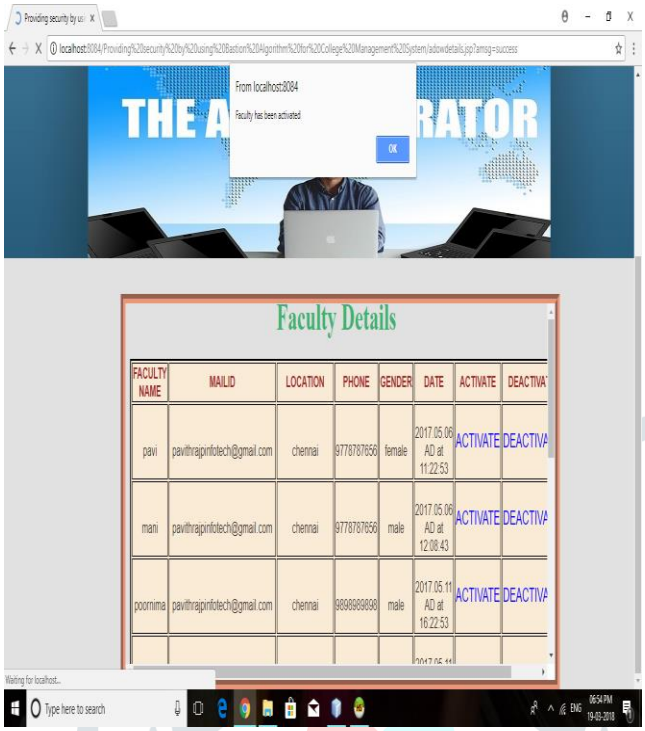
There are different types of modules in the system architecture of college management system. They are as follows:
•Login
•Data Upload
•Key Management
•Data Search & View

Faculty and students has to register in to cloud. For registration, faculty and students has to fill the details such as name, email-id, mobile number, gender etc.. Faculty and students has to login in to cloud to view the file and to download the file that is present in the cloud. By using signature key and private key, faculty and students can view the login details i.e., details of their profile.Faculty will upload data in to cloud. Data such as materials, textbooks, circulars etc can be uploaded to cloud. Different faculty members can upload different types of data.In key management, there are two sub modules, they are:Key request,Key response.Students will send the search key request to college administrator. Students will send this key to access data that is present in cloud.Administrator will send the search key response to students. By using this key, students can access the data present in cloud.Students can search all the data that is uploaded by all faculty members. After entering the search key, students can view the data present in cloud. Faculty can also view the data that are uploaded by other faculty members. Students and faculty members can also download the data present in cloud.

VI.EXPERIMENTAL RESULTS


POLYNOMIAL TIME KEY
SUBMIT
KEY_1
KEY_2
KEY_3
KEY_4
KEY_5
KEY_6
Keygenerate_1
Keygenerate_2
Keygenerate_3
Keygenerate_4
Keygenerate_5
Keygenerate_6


priyaa softwarelink java 2017.05.13 AD at 17:34:27 Generated KEYGENERATE view Download
habeeb abc liy 2018.03.02 AD at 11:07:08 Generated KEYGENERATE view Download
saiteja def file1 2018.03.15 AD at 10:31:42 Generated KEYGENERATE view Download
habeeb hello abc 2018.03.19 AD at 18:49:33 keygenerate KEYGENERATE view Download



VII.CONCLUSION & FUTURE ENHANCEMENTS

We addressed the problem of securing data outsourced to the cloud against an adversary which has access to the encryption key. For that purpose, we introduced a novel security definition that captures data confidentiality against the new adversary. We then implement Bastion, a scheme which ensures the confidentiality of encrypted data even when the adversary has the encryption key, and all but two ciphertext blocks. Bastion is most suitable for settings where the ciphertext blocks are stored in multi-cloud storage systems. In these settings, the adversary would need to acquire the encryption key, and to compromise all servers, in order to recover any single block of plaintext. Encryption can be done by using polynomial key generation.

VII. REFERENCES

[1] M. Abd-El-Malek, G. R. Ganger, G. R. Goodson, M. K. Reiter, and J. J. Wylie, "Fault-Scalable Byzantine Fault-Tolerant Services," in ACM Symposium on Operating Systems Principles (SOSP), 2005, pp. 59–74.
[2] M. K. Aguilera, R. Janakiraman, and L. Xu, "Using Erasure Codes Efficiently for Storage in a Distributed System," in International Conference on Dependable Systems and Networks (DSN), 2005, pp. 336–345.
[3] W. Aiello, M. Bellare, G. D. Crescenzo, and R. Venkatesan, "Security amplification by composition: The case of doublyiterated, ideal ciphers," in Advances in Cryptology (CRYPTO), 1998, pp. 390–407.
[4] C. Basescu, C. Cachin, I. Eyal, R. Haas, and M. Vukolic, "Robust Data Sharing with Key-value Stores," in ACM SIGACTSIGOPS Symposium on Principles of Distributed Computing (PODC), 2011, pp. 221–222.
[5] A. Beimel, "Secret-sharing schemes: A survey," in International Workshop on Coding and Cryptology (IWCC), 2011, pp. 11–46.
[6] A. Bessani, M. Correia, B. Quaresma, F. André, and P. Sousa, "DepSky: Dependable and Secure Storage in a Cloud-ofclouds," in Sixth Conference on Computer Systems (EuroSys), 2011, pp. 31–46.
[7] G. R. Blakley and C. Meadows, "Security of ramp schemes," in Advances in Cryptology (CRYPTO), 1984, pp. 242–268.

[8] V. Boyko, "On the Security Properties of OAEP as an Allor-nothing Transform," in Advances in Cryptology (CRYPTO), 1999, pp. 503–518.
[9] R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky, "Deniable Encryption," in Proceedings of CRYPTO, 1997.
[10] Cavalry, "Encryption Engine Dongle," http://www. cavalrystorage.com/en2010.aspx/.
[11] C. Charnes, J. Pieprzyk, and R. Safavi-Naini, "Conditionally secure secret sharing schemes with disenrollment capability," in ACM Conference on Computer and Communications Security (CCS), 1994, pp. 89–95.
[12] A. Desai, "The security of all-or-nothing encryption: Protecting against exhaustive key search," in Advances in Cryptology (CRYPTO), 2000, pp. 359–375.
[13] C. Dubnicki, L. Gryz, L. Heldt, M. Kaczmarczyk, W. Kilian, P. Strzelczak, J. Szczepkowski, C. Ungureanu, and M. Welnicki, "HYDRAstor: a Scalable Secondary Storage," in USENIX Conference on File and Storage Technologies (FAST), 2009, pp. 197–210.
[14] M. Dürmuth and D. M. Freeman, "Deniable encryption with negligible detection probability: An interactive construction," in EUROCRYPT, 2011, pp. 610–626.
[15] EMC, "Transform to a Hybrid Cloud," http://www.emc. com/campaign/global/hybridcloud/index.htm.
[16] IBM, "IBM Hybrid Cloud Solution," http://www-01.ibm. com/software/tivoli/products/hybrid-cloud/.
[17] J. Kilian and P. Rogaway, "How to protect DES against exhaustive key search," in Advances in Cryptology (CRYPTO), 1996, pp. 252–267.
[18] M. Klonowski, P. Kubiak, and M. Kutylowski, "Practical Deniable Encryption," in Theory and Practice of Computer Science (SOFSEM), 2008, pp. 599–609.
[19] H. Krawczyk, "Secret Sharing Made Short," in Advances in Cryptology (CRYPTO), 1993, pp. 136–146.
[20] J. Kubiatowicz, D. Bindel, Y. Chen, S. E. Czerwinski, P. R. Eaton, D. Geels, R. Gummadi, S. C. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Y. Zhao, "OceanStore: An Architecture for Global-Scale Persistent Storage," in International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2000, pp. 190–201.