# Ensuring User Security and Data Integrity in Multi-Cloud

**Kavitha Murugesan, Shilpa Sudheendran**

***Abstract - The most common method used for user security is textual passwords. But they are vulnerable to eves dropping, dictionary attack, shoulder surfing etc. To address this problem, text can be combined with colors to generate session passwords called hybrid textual authentication for authentication in multi – cloud. Provable data possession is a technique for ensuring data integrity. We present a cooperative provable data possession scheme based on homomorphic verifiable response and hash index hierarchy for ensuring integrity of data in cloud storage.***

***Keywords- Session passwords, Hybrid textual authentication, Multi-Cloud, Provable data possession, Cooperative provable data possession.***

## I. INTRODUCTION

Cloud computing has become a faster profit growth point in recent years by providing a comparably scalable, low cost, position-independent platform for data outsourcing. Although commercial cloud services have evolved around public clouds, the growing interest of building private cloud on open source cloud computing tools forces local users to have a flexible and agile private infrastructure to run service workloads within their administrative domains. Private clouds are not exclusive for being public clouds, and they can also support a multi-cloud model by supplementing a local infrastructure with computing capacity from external public clouds. A multi- cloud allow remote access to its resources over the Internet via remote interfaces, by using virtual infrastructure management (VIM) [1], such as the Web service interfaces that Amazon EC2 uses.

There are different tools and technologies that emerged recently for multi-clouds such as the Platform VM Orchestrator, VMware VMsphere, and Ovirt. They help users construct a comparably scalable, low-cost, location-independent platform for managing client's data. If such a platform is vulnerable to security attacks, it would bring irrevocable losses to the clients. The confidential data in an enterprise may be illegally accessed by using remote interfaces, or the relevant data and archives are lost or tampered with when they are stored into an uncertain storage pool outside the enterprise. Therefore, it is crucial for cloud service providers (CSPs) to provide secure management techniques to ensure their storage services.

The most popular user authentication approach is the text-based password scheme in which a user enters a login name and password. Despite of its wide usage, the textual passwords have a number of short comes.

**Manuscript received May, 2013**.

**Kavitha Murugesan**, Computer Science, Vedavyasa Institute of Technology, Calicut, India.

**Shilpa Sudheendran**, Computer Science, Vedavyasa Institute of Technology, Thrissur, India.

The simple and straightforward textual passwords are easy to remember, but they are more vulnerable for attackers to break. Whereas the complex and arbitrary passwords makes the system more secure, resisting the brute force search and dictionary attacks, but the difficulty lies in retaining them [2]. Besides this, textual passwords are liable to the shoulder-surfing, hidden cameras, and spyware attacks.

Consequently, graphical password strategies have been introduced as a substitute for textual passwords schemes, as pictures can be easily remembered when compared with words. Furthermore, it is difficult to formulate automated attacks for graphical passwords. Moreover the password space of the graphical password scheme may extend that of the textual based password schemes and hence probably providing a higher level of security [3]. On the account of these reasons, there is an intensifying interest in the graphical password methods.

However, most of the existing graphical password authentications methods suffer from shoulder surfing, a known hazard where an intruder can scrutinize the password by recording the authentication session or through direct surveillance. In addition, setting up a system that offers the graphical authentication schemes is substantially costlier than the text based password methods. Even though some of the graphical password procedures resistant to the shoulder surfing are proposed, yet they have their own downside like usability issues or consuming additional time for user to login or having tolerance levels [4].

Based on these various reasons pointed out, session passwords are instigated. Session passwords are those that can be used only at that particular instant. As soon as the session expires, the password is no longer valid. As such the user, keys in distinct passwords each time he logs into the session.

Provable data possession (PDP) [5] is a probabilistic proof technique for a storage provider to prove that client's data remains intact or the clients can fully recover their data and have confidence to use the recovered data. This creates strong demand for seeking an effective solution to check if their data has been tampered with or deleted without downloading the latest version of data. Various PDP schemes have been recently proposed, such as Scalable PDP [6] and Dynamic PDP [7], to work in a publicly verifiable way so that users can employ their verification protocols to prove the availability of the stored data. However, these schemes focus on the PDP issues at untrusted servers (public clouds), and are not applicable for a multi-cloud environment.

In this paper, our objective is to provide security to the confidential files residing in multi-cloud. Here, any user who needs to the access those files has to first get registered. The registered user is then allowed to login using session passwords through the hybrid textual authentication technique. If the user is ascertained as the genuine person

then he is given the rights to access the confidential files or else he is regarded as unauthorised.We address the problem of provable data possession in multi-clouds. By discussing the characteristics of multi-clouds and analyzing security drawbacks of the existing schemes, we point out our main research objectives in three aspects: high performance, high security and verification transparency. On this basis, we propose a verification framework for multi-clouds along with the main techniques adopted in our approach: (1) fragment structure, (2) hash index hierarchy (HIH), and (3) homomorphic verifiable response (HVR).

## II. RELATED WORKS

Dhamija and Perrig[2] proposed a new graphical authentication scheme. In this method, while creating the password allows the user to select certain number of pictures from a set of random images. Then, during login, the user has to recognize the preselected portraits from the set of images. But this method is liable to shoulder-surfing.

Blonder [8] proposed a graphical password technique, in which the password is generated by allowing the user to click on different positions on an image. During authentication, the user has to click on the estimated areas of those locations. Later, this idea was prolonged by 'pass-point system' where the predefined boundaries are excluded and arbitrary images are supported. Consequently, for constructing password, the user can click over any region on the image. A tolerance around each chosen pixel is evaluated. To be authenticated, the user has to click within the tolerance level of the pixels chosen.

Syukri [9] designed a scheme in which authentication is carried out by sketching out the user signature with mouse. This scheme involves two levels, registration and verification. While registering, the user draws his signature using mouse, the system then extracts the signature area. During the verification level, it acquires the user signature as input, performs normalization and finally extracts the parameters of the signature. But this scheme is associated with several disadvantages such as forgery of signatures, inconvenience while drawing with mouse, difficulty in sketching the signature in the same perimeters at the time of registration.

Passface[10] is an approach proposed by the Real User Corporation in which the user is allowed to choose four images of human faces from the face database as their password. During the verification phase, the user is provided with a grid of nine faces, one already chosen during the registration and eight decoy faces. The user identifies the selected face and clicks anywhere over it. This course of action is repeated for four times, and the user is ascertained as genuine if he recognizes all faces accurately.

Jansen [11, 12] proposed an innovative authentication scheme for mobile devices. While creating the password, the user chooses a theme of snapshots in thumbnail size and the sequence of those snapshots is fixed as password. As thumb nail is associated with numerical value, the sequence of image form numerical password. The only drawback with this method is that the password space is not large, as no of images is limited to 30.

Man, et.al [13] proposed a technique to overcome shoulder-surfing. In this system, the user selects many portraits as pass objects. Each pass object is allotted a unique code. During verification process, the user has to input those unique codes of the pass objects in the login interfaces presented by the system. This scheme resists the hidden camera; but the user has to memorize all pass object codes.

Juels, et.al [14] proposed a new cryptographic building block known as a proof of retrievability(POR). A POR enables a user (verifier) to determine that an archive (prover) "possesses" a file or data object F. More precisely, a successfully executed POR assures a verifier that the prover presents a protocol interface through which the verifier can retrieve F in its entirety. Of course, a prover can refuse to release F even after successfully participating in a POR. A POR, however, provides the strongest possible assurance of file retrievability barring changes in prover behavior.

Shacham, et.al [15] proposed the first proof-of-retrievability schemes with full proofs of security. Their first scheme, built from BLS signatures and secure in the random oracle model has the shortest query and response of any proof-of-retrievability with public verifiability. Their second scheme, which builds elegantly on pseudorandom functions (PRFs) and is secure in the standard model, has shortest response of any proof-of-retrievability scheme with private verifiability but a longer query. Above two schemes rely on homomorphic properties to aggregate a proof into one small authenticator value.

Ateniese et.al [6] constructed a highly efficient and provably secure PDP technique based entirely on symmetric key cryptography. Also, our PDP technique allows outsourcing of dynamic data, it efficiently supports operations, such as block modification, append and deletion.

Ateniese et.al [5] introduced a model for provable data possession (PDP) that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving

it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which significantly reduces I/O costs. A constant amount of metadata to verify the proof is maintained by the client. The challenge/response protocol transmits a constant, small amount of data, which minimizes network communication. The PDP model for remote data checking supports large data sets in widely-distributed storage systems.

Erway et.al [7] constructed a dynamic provable data possession (DPDP), which extends the PDP model to support provable updates on the stored data. DPDP solution is based on a new variant of authenticated dictionaries which use rank information to organize dictionary entries. Thus we can support efficient authenticated operations on files at block level, such as authenticated insert and delete. They proved the security of their constructions using standard assumptions.

## III. PROPOSED SOLUTION

In this section we present a framework for multi-cloud, explanation about hybrid textual authentication [16] and cooperative provable data possession (CPDP) [17].CPDP includes two techniques hash index hierarchy and homomorphic verifiable response.

### *A. Hybrid Textual Authentication*

Authentication technique consists of three phases: registration phase, login phase and verification phase. In registration phase, user rates the colors. In login phase, the user has to enter the password based on the interface displayed on the screen. Then the system verifies the

password entered by comparing with content of the password generated during registration.

During registration, user should rate colors as shown in fig.1. The User should rate colors from 1 to 8 and he can remember it as "RBYOLGEP". Same rating can be given to different colors. During login, when the user enters his username an interface is displayed based on the colors selected by the user. The login interface consists of an 8×8 grid. This grid contains digits 1-8 placed randomly in grid cells as shown in fig.2. The interface also contains a color grid which consists of 4 pairs of colors and each pair of color represents the row and the column of the grid.


Fig.1. Color rating by user

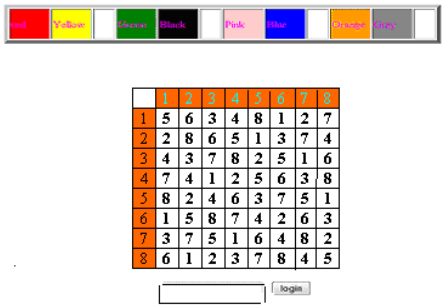


Fig.2. Login interface

The number grid of 8 x 8 has numbers 1 to 8 randomly placed in the grid. We get the session password depending on the ratings given to colors, i.e. the first color of every pair in color grid represents row and second represents column of the number grid. The number at the intersection of the row and column of the grid is part of the session password. Consider the figure 1 ratings and figure 2 login interface for demonstration. The first pair of color grid has red and yellow colors. The rating of red color is 1 and yellow color is 3. So the first letter of session password is 1st row and 3rd column intersecting element is **3**. The same method is followed for other pairs of colors. For figure 2 the password is "3573**"**. Instead of digits, alphabets can be used. For each login, both the number grid and the color grid get randomizes so the session password changes for every session.

### *B. Framework for Data Integrity in Multi-cloud*

In this architecture, we consider a data storage service involving three different entities: Granted clients, who have a large amount of data to be stored in multi-clouds and have the permissions to access and manipulate the stored data; Cloud service providers (CSPs), who work together to provide data storage services and have enough storage space sand computation resources; and Trusted third parties (TTPs),who are trusted to store the verification parameters and offer the query services for these parameters. Fig.3 shows the architecture.

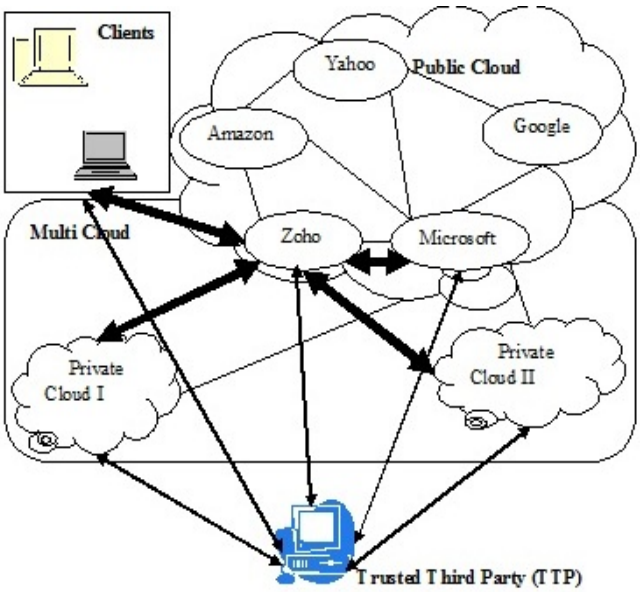


Fig.3. Framework for data integrity in multi-cloud

In this architecture, we consider the existence of multiple CSPs to collaboratively store and maintain the client's data. Moreover, a cooperative PDP is used to verify the integrity and availability of stored data in CSPs. The verification procedure is described as follows: Firstly, the client (data owner) uses the secret key to pre-process the file, which consists of a collection of n blocks. It generates a set of public verification information that is stored in TTP, then transmits the file and some verification tags to CSPs, and may delete its local copy. Later by using a verification protocol for cooperative PDP, the clients can issue a challenge for one CSP to check the integrity and availability of outsourced data in terms of public verification information stored in TTP.

### *C. Definition of Cooperative PDP*

A cooperative provable data possession scheme S′ is a collection of two algorithms and an interactive proof system, S′ = (K, T, P).

KeyGen**(**$1^k$**):** It takes a security parameter k as input, and returns a secret key sk or a public-secret key pair (pk, sk);

TagGen**(**sk, F,P**):** It takes as inputs a secret key sk, a file F, and a set of cloud storage providers $\mathrm{P} = \{\mathrm{P_k}\}$, and returns the triples (ζ ,ψ, σ), where ζ is the secret of tags, ψ = (u,H) is a set of verification parameters u and an index hierarchy H for F, σ = $\{\sigma^{(k)}\}\mathrm{P_k}\in \mathrm{P}$ denotes a set of all tags, $\sigma^{(k)}$ is the tags of the fraction $\mathrm{F}^{(k)}$ of F in $\mathrm{P_k}$.

Proof**(**P, V **):** It is a protocol of proof of data possession between the CSPs (P = $\{\mathrm{P_k}\}$) and a verifier (V), that is,$(\sum_{\mathrm{Pk}\in \mathrm{P}}\mathrm{P_k}(\mathrm{F}^{(k)}, \sigma^{(k)}), \mathrm{V})$ (pk, ψ), where each $\mathrm{P_k}$ takes as input a file $\mathrm{F}^{(k)}$ and a set of tags $\sigma^{(k)}$, and a public key pk and a set of public parameters ψ is the common input between P and V . At the end of the protocol run, V returns a bit {0|1} denoting false and true where, $\sum_{\mathrm{Pk}\in \mathrm{P}}$ denotes the collaborative computing in $\mathrm{P_k} \in \mathrm{P}$.

### *D. Hash Index Hierarchy*

A representative architecture for data storage in multi-clouds is illustrated as follows: this architecture is a hierarchical structure H on three layers to represent the

relationships among all blocks for stored resources. Three layers are as follows:

- First-Layer (*Express Layer*): offers an abstract representation of the stored resources;
- Second-Layer (*Service Layer*): immediately offers and manages cloud storage services;
- Third-Layer (*Storage Layer*): practicably realizes data storage on many physical devices;

This kind of architecture is a nature representation of file storage. We make use of this simple hierarchy to organize multiple CSP services, which involve private clouds or public clouds, by shading the differences between these clouds. In this architecture, the resources in Express Layer are split and stored into three CSPs in Service Layer. Each CSP fragments and stores the assigned data into the storage servers in Storage Layer. We follow the logical order of the data blocks to organize the Storage Layer. Moreover, this architecture could provide some special functions for data storage and management. For example, there may exist overlap among data blocks and skipping. But these functions would increase the complexity of storage management.

### *E. Homomorphic Verifiable Response*

A homomorphism is a map $f : P \rightarrow Q$ between two groups such that $f(g1+g2) = f(g1)\times f(g2)$ for all $g1, g2 \in P$, where + denotes the operation in P and × denotes the operation in Q. This notation is used to define Homomorphic Verifiable Tags (HVTs): Given two values $\sigma_i$ and $\sigma_j$ for two message $m_i$ and $m_j$, anyone can combine them into a value $\sigma'$ corresponding to the sum of the message $m_i + m_j$.

## IV. CONCLUSION

In this paper we used hybrid textual authentication for ensuring user security in multi-cloud and cooperative provable data possession based on techniques hash index hierarchy and homomorphic verifiable response used for data integrity verification. Authentication scheme protected us from shoulder- surfing, dictionary attacks. CPDP scheme provided security and integrity to data stored on cloud.

## REFERENCES


[1] B. Sotomayor, R.S. Montero, I.M. Llorente, and I.T. Foster, "Virtual Infrastructure Management in Private and Hybrid Clouds," IEEE Internet Computing, vol. 13, no. 5, pp. 14-22, Sept. 2009.

[2] R. Dhamija, and A. Perrig. "DéjàVu: A User Study Using Images for Authentication". In 9th USENIX Security Symposium, 2000.

[3] H. Zhao and X. Li, "S3PAS: A Scalable Shoulder-Surfing Resistant Textual-Graphical Password Authentication Scheme," in 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW 07), vol. 2. Canada, 2007, pp. 467-472.

[4] Haichang Gao, Zhongjie Ren, Xiuling Chang, Xiyang Liu Uwe Aickelin, "A New Graphical Password Scheme Resistant to Shoulder-Surfing", 2010 International Conference on CyberWorlds, 20-22 October 2010, Singapore.

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

[6] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. Fourth International Conf. Security and Privacy in Comm. Netowrks (SecureComm '08), pp. 1-10, 2008.

[7] C.C. Erway, A. Kupcu , C. Papamanthou, and R. Tamassia, "Dynamic Provable Data Possession," Proc. 16th ACM Conf. Computer and Comm. Security (CCS '09), pp. 213-222, 2009.

[8] G. E. Blonder. Graphical passwords. *United States Patent 5559961*, 1996.

[9] A. F. Syukri, E. Okamoto, and M. Mambo, "A User Identification System Using Signature Written with Mouse," in Third Australasian Conference on Information Security and Privacy (ACISP): Springer-Verlag Lecture Notes in Computer Science (1438), 1998, pp. 403-441.

[10] Real User Corporation: Passfaces. www.passfaces.com

[11] W. Jansen, "Authenticating Mobile Device User through Image Selection," in Data Security, 2004.

[12] W. Jansen, "Authenticating Users on Handheld Devices" in Proceedings of Canadian Information Technology Security Symposium, 2003.

[13] S. Man, D. Hong, and M. Mathews, "A shoulder surfing resistant graphical password scheme," in Proceedings of International conference on security and management. Las Vegas, NV, 2003

[14] A. Juels and B.S.K. Jr., "Pors: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 584-597, 2007.

[15] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '08), pp. 90-107, 2008.

[16] K.Nivetha, M. Muthumeena, R. Srinivasan ," Authentication Mechanisim For Session Passwords By Imposing Color With Text ", International Journal of Engineering Research and Applications (IJERA) Vol. 2, Issue 5, September- October 2012, pp.1611-1615

[17] Yan Zhu, Hongxin Hu,Gail-Joon Ahn, Mengyang Yu, "Cooperative Provable Data Possession for Integrity Verification in Multicloud Storage", IEEE Transactions On Parallel And Distributed Systems, Vol. 23, No. 12, December 2012