Strong Authentication for Cloud Computing using False OTP

Ankit Baghel[1], Deepak Agarwal[2]
[1] Research Scholar Dept of CSE
[2] Assistant Professor Dept of CSE
[1, 2] Takshila Institute of Engineering & Technology
Jabalpur, Madhya Pradesh, India.

Abstract- *Modern Authentication system ensures that only the entrusted entities should communicate and ensures that unauthorized user which is trying to establish some connection in order to gain login to the system by impersonating his identity should be restricted to the confidential data. Now days, modern authentication system receives more and more attention as the data exchanged over the network is important and confidential. So it is more likely to be vulnerable to the known security threats. However the design of the secure authentication protocol is quite challenging, considering the known security threats. Therefore, we suggest a new authentication and integration framework to secure data and information hacks.*

User authentication in proposed work is performed on the basis of secure OTP & user name and email password. It is verified on the basis of several security aspects and is verified to be available, accessible, feasible, secure, and user-friendly and provides strong authentication system. The proposed framework shows the close agreement with the standard criteria for security.

Keywords- Security, user authentication, False OTP, Cloud Computing, Hinglish, Shoulder-Surfing.

I. INTRODUCTION

Cloud computing implements virtualization technique is to provide resources efficiently to the end user. The characteristics of cloud computing include manageability, scalability, and availability. In addition, cloud computing is also economical, on-demand service, expedient, ubiquitous, multitenant, elasticity, and stability. Cloud computing offers mainly three service delivery models; Infrastructure as a Service (IaaS), Platform as a Services (PaaS) and Software as a Service (SaaS) [1]. NIST defines four-development model of the cloud: public, private, hybrid and community. Cloud computing uses cloud server stack where the client or user is on the front end and server on the back end. Services reside in middleware of stack as shown in Fig.1. At the top level resides the application, which directly delivers the outsourced software to the client and eliminates sophisticated software. Customers do no need to expend money to install software, only they pay for their usage [2]. In the last few decades, Internet technology has advanced a lot. Cloud computing is an internet based ubiquitous, on demand network model for convenient network access (e.g. Servers, Applications, Services and Networks) for pool of configurable computing resources as on demand basis. The software & data that is accessed by the user or a customer may be stored in different servers at different geographical places. This is a security challenge for both the service providers and users. Different organizations utilize the cloud as per the convince based on different service models (SaaS, PaaS & IaaS) and four deployment models (Public cloud, Hybrid cloud, Private cloud and Community cloud). There are a number of security issues and concerns which are to be resolved in cloud computing. All these issues fall into two broader categories i.e. security issues faced by cloud providers and security issues faced by their customers. Both have different set of responsibilities from encryption, strong password and authentication measures to fortification. In this emerging technology users can deal with a service without any clue of where the actual infrastructure is located and what technology is used behind the scenes to manage and control the infrastructure [3].

1.2 Cloud Computing Security Issues and Threats

Cloud computing offers services using the Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) models [4]. Cloud users have access to servers and virtual machines through the IaaS service model. The hypervisors execute on the servers to provide virtualization of physical resources. Similarly, using the PaaS service model, the cloud platform provides support of operating systems, runtime systems, databases or web servers. The SaaS service model provides support of pay-per-use software. The diversity of these service delivery models makes the cloud computing platforms more vulnerable to attacks than any other computing platform. Its vulnerability may be exposed through any of its core components: network, virtual machines, storage and applications, which are used as a basis for categorization of attacks and their implications.

1.3 User Authentication in the Cloud

Based on Cloud Service Model, security issues can be categorized [5]. It can be categorized into network level, user authentication level, data level, and generic issues. Each cloud service model comprises its own inherent security flaws; however they also share some challenges that affect all of them. Before analyzing security challenges in Cloud Computing, we need to understand the relationships and dependencies between these cloud service models [6]. PaaS as well as SaaS are hosted on top of IaaS thus, any breach in IaaS will impact the security of both PaaS and SaaS services, but also it may be true on the other way around. However, we have to take into account that PaaS offers a platform to build and deploy SaaS applications, which increases the security dependency between them. As a consequence of these deep dependencies, any attack to any cloud service layer can compromise the upper layers. These relationships and dependencies between cloud models may also be a source of security risks. A SaaS provider may rent a development environment from a PaaS provider, which might also rent an infrastructure from an IaaS provider. Each provider is responsible for securing his own services, which may result in an inconsistent combination of security models. It also creates confusion over which service provider is responsible once an attack happens. The Dependencies between three services is shown in figure 1.3.

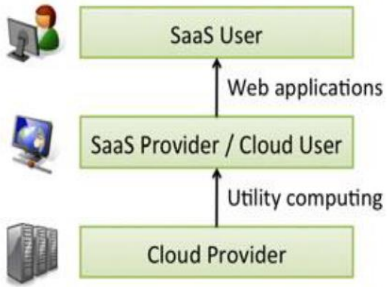

Figure 1.1: Cloud service broker model.

II. LITERATURE SURVEY

2.1 Existing Cloud Authentication Mechanisms

This section discusses some authentication mechanisms which are used by various web applications like cloud. It also introduces advantages and disadvantages of every authentication scheme. Most of the application use username-password methods for authentication. There are various password cracking tools, which are available free online. Hence attacker takes few minutes for cracking the password [7]. For keeping safety from this type of attacks National Institute for Standards and Technology (NIST) and Federal Financial Institutions Examination Council (FFIEC) gives some instruction to carry out financial transactions. Some of the works found in paper [8] about two tier authentications or two factor authentication mechanisms.

As the researchers discuss the three mentioned categories, they will also analyze different kinds of each pertinent model. To clarify more, the username and password that are regularly used on websites are in the first group of authentication model whereas the credit card is counted as the ownership factor to check the validity during an authentication process. The last model includes biometric features with the capability of proving who the system users are. Recently many security researchers are focusing on various new techniques of authentication in cloud computing that include one or more of the above mentioned methods of authentication. Therefore it becomes inevitable to survey the various authentication methods recently proposed and implemented in the Cloud computing environment. The distribution of different Authentication method in Cloud computing is shown in figure

2.1.

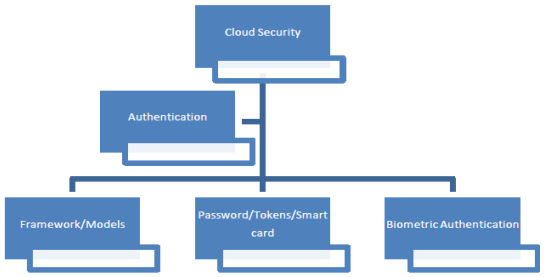

Figure 2.2: Different Authentication method in Cloud computing.

2.2 OTP Generation Mechanism

A one-time password (OTP) is a password that is valid for only one login session or transaction. OTPs avoid a number of shortcomings that are associated with traditional (static) passwords. The most important shortcoming that is addressed by OTPs is that, in contrast to static passwords, they are not vulnerable to replay attacks. This means that a potential intruder who manages to record an OTP that was already used to log into a service or to conduct a transaction will not be able to abuse it, since it will be no longer valid. On the downside, OTPs are difficult for human beings to memorize. Therefore they require additional technology to work.

2.3 How to generate OTP and distribute?

OTP generation algorithms typically make use of pseudo randomness or randomness. This is necessary because otherwise it would be easy to predict future OTPs by observing previous ones. Concrete OTP algorithms vary greatly in their

details. Various approaches for the generation of OTPs are listed below:

Based on time-synchronization between the authentication server and the client providing the password. (OTPs are valid only for a short period of time).

Using a mathematical algorithm to generate a new password based on the previous password (OTPs are effectively a chain and must be used in a predefined order).

Using a mathematical algorithm where the new password is based on a challenge (e.g., a random number chosen by the authentication server or transaction details) and/or a counter.

There are also different ways to make the user aware of the next OTP to use. Some systems use special electronic security tokens that the user carries and that generate OTPs and show them using a small display. Other systems consist of software that runs on the user's mobile phone. Yet other systems generate OTPs on the server-side and send them to the user using an out-of-band channel such as SMS messaging. Finally, in some systems, OTPs are printed on paper that the user is required to carry.

2.4 Existing Method for Generating OTP in Cloud Computing

In paper [9] introduce a trusted proxy for cloud services such as Dropbox, to handle authentication using our novel challenge-response protocol that is based on prime number factorization. The system is first set-up by requiring the client to authenticate itself to the cloud service using the normal user-password authentication protocol, and then requests for an access token. It then authorizes the trusted proxy to mediate the authentication between him and the cloud service, by passing the access token to the trusted proxy.

In this one-time set-up phase, two secrets, which are randomly generated large prime numbers, are distributed to the client, together with a client secret, to be stored in a secure storage. A public composite number, which is the product of the two prime numbers, is stored in the trusted proxy to identify the client. Knowing this composite number does not reveal any information about the secret prime numbers. Once this proxy-based cloud authentication has been setup, all subsequent authentication will not require the user to transmit its username-password, but a security challenge will be issued to the client based on the composite number, and a residue computed based on the client's secret. Only the correct response which must be generated using the client's secrets will be accepted. The challenge is generated using a random number that is 3072 bits long, and hence ensuring that each access request is unique and will not be re-used. The database is used to store the mapping of the public composite number to the corresponding user account. Using this protocol, the clients do not need to use their username-password to perform authentication anymore.

2.5 LIMITATIONS OF EXISTING TECHNIQUES

In the literature review section, we discuss various authentication techniques along with their advantages and disadvantages. The disadvantages of these techniques can be divided into four parameters namely security from insider attacks, presence of authentication control towards client or server, extra software and hardware needed and number of security tiers required. A brief summary of the comparison based upon these parameters is presented below:

1) Security from Insider Attacks: This parameter is based on the assumption that it is easy for an insider to gain access to first-tier login and password. This is not adoptable. So, multitier authentication is also required.
2) Presence of Authentication Control towards Client or Server: In cloud computing environment, people store their data on the remote storage cloud (third party storage) and this data is controlled and maintained by third party service provider. So, there should be some technique for authentication of user at client side.

III. PROPOSED SYSTEM

3.1 Proposed Model

Figure shown below, represents working of the proposed system. Proposed system is a web application hosted on a cloud server. That means proposed system will be on a web server hosted by a cloud service provider. Proposed system generates more secure one time password for user authentication. System will perform multifactor authentication using proposed OTP generation method. Normally OTP is numbered OTP. Proposed system will generate numbered OTP which is false and a 10 character hinglish key based on hinglish character set. User gets numbered OTP which should be converted into actual hinglish OTP by index mapping and entered at the time of login.

User requests system to generate One Time Password. Then, a unique key for every user will be generated from predefined hinglish character set. Proposed system uses English and Hindi language character set to produce unique key. That will enhance randomness of key and reduces chances to crack it. Keys will be generated by randomly selecting character for

digits from zero to nine. Means key is a ten character key containing symbols from english, hindi and numbers. Than a numeric six digit time based false OTP will be generated. An email consists of both hinglish key and numeric OTP will be delivered to users registered email id.

For OTP authentication, user should read email containing hinglish key and false OTP. Then user should convert numbered false OTP into actual OTP by taking characters from key at corresponding index. For user friendliness system will accept OTP using system generated hinglish virtual key board, which contains only symbols from user keys in random order. OTP submitted will be matched with actual OTP that will be generated by the system. If OTP is correct than user is authorized to use resources available on cloud.

3.2 Proposed method of hinglish key generation algorithm

In proposed system fake numbered OTP for every user will be generated according to random number generating theory. Actual OTP will be generated from a hinglish key that is unique key for every user generated by proposed system. System proposes a new key generation method, which generates key string using english alphabets, hindi alphabets and some special characters. Alphabets are taking from different languages because that increases randomness in generating key which will be difficult to do brute force attack. Key will be mailed on registered mail id of the user. This key will be utilized by user for encoding the numbered false OTP received in email to actual hinglish OTP. Random numbers will be generated for digit 0 to 9 between 0 to 59. Corresponding symbols to random number will be stored in a array for digits of size 10. After this all symbols corresponding to digits are concatenated to make key string.

Predefined collection of hinglish symbols utilized to generate key string is shown in figure below:

द	d	@	f	व	स	z	u	ल	r
0	1	2	3	4	5	6	7	8	9

So generated hinglish key string is द d@ f व स z u ल r

Algorithm contains following steps:
Step 1: New user should register in the system by giving information like user name, password, email id, mobile number etc.
Step 2: This information will be checked against already registered user than saved in database.
Step 3: A predefined array t of size 60 will be taken, which contains alphabets of hindi, english & some other symbols. , ; ? ! : &.
Step 4: loop for digits d=0 to 9
Step 4.1: a method random will generate random number between 0 to 59 which is corresponding to index of predefined array t, suppose it is i
Step 4.2: store k[d] =t[i]
Step 5: Store key string k in database.
Step 6: Send key string k on user's registered email id.

IV. PERFORMANCE EVALUATION

Above snapshot will be displayed on clicking of login button in home page. It will ask for text based authentication known as one factor authentication. User should enter registered email id and password, then system will match it with stored password and on authenticated user system will display OTP page for more secure authentication otherwise again enter login details. For evaluating the proposed system with existing OTP system login page has two buttons-one for existing OTP and another for proposed OTP. Existing system based OTP will be sent to registered email id directly on clicking of existing log in button. While for proposed system OTP will be generated after log in on android application in smart phone. Figure 4.1 below represents received OTP in email.

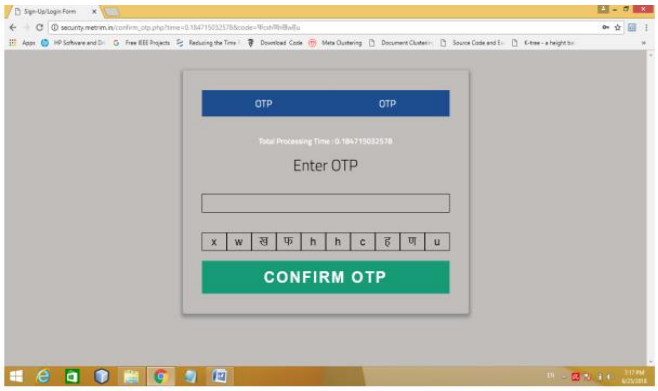

Figure 4.1: OTP in email.

After receiving OTP and key user will submit OTP for authentication on figure 4.2 below:

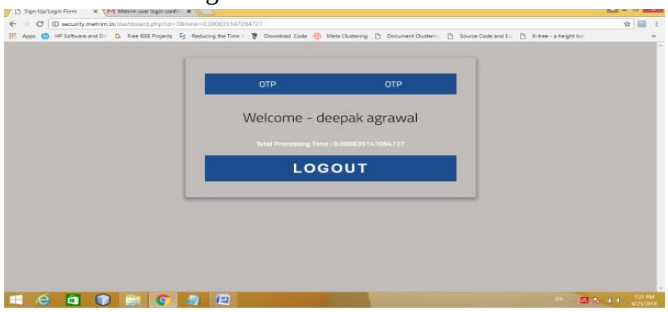

Figure 4.2: OTP Authentication module.

V RESULTS & EVALUATION

In this section, we give a security analysis and a performance analysis of the proposed system. Proposed system is more efficient and secure because of following reasons: Out-of-band authentication is becoming more popular given that customer PCs are increasingly vulnerable to malware attacks. Out-of-band authentication means that a transaction that is initiated via one delivery channel (e.g., Internet) must be re-authenticated or verified via an independent delivery channel (e.g., mobile). Fake OTP concept also provide more security because if attacker got succeed then he will got fake OTP which can never be validated. User should need android application for finding actual OTP. Therefore, replay attack is infeasible in the proposed system. In proposed system actual OTP is generated by predefined symbol set consist of hindi and English alphabets with some special symbols which reduces to crack it. Proposed scheme is resistant to man-in-the-middle attack. User got fake OTP. So if someone fetches it then it will not be authenticated. Therefore, the man-in-the-middle attack is infeasible for the proposed system.

Table 5.1 Evaluation on the basis of total OTP generation time.

Method	Time(in microseconds)		
	Attempt 1	Attempt 2	Attempt 3
OTP with existing method	1.211	1.176	0.172
Proposed method	0.817	1.161	0.151

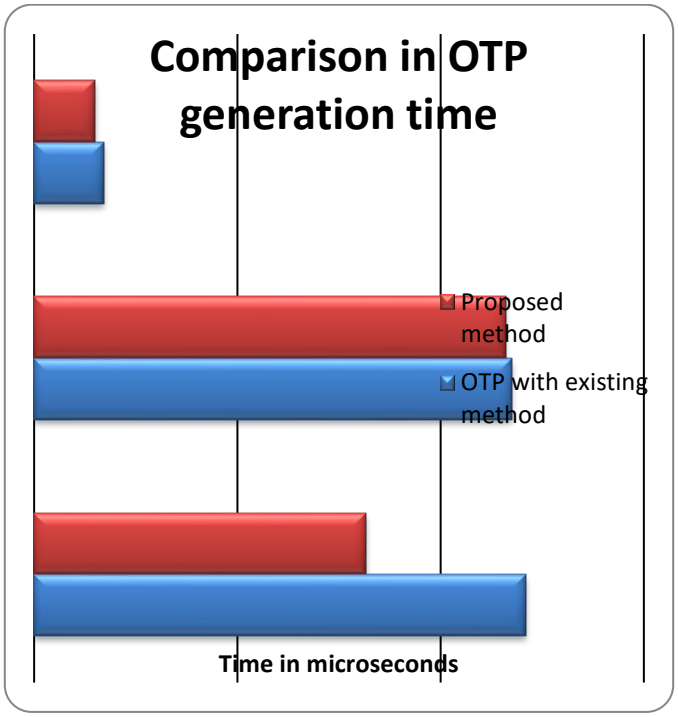

Figure 5.2: Total OTP generation time comparison chart

Chart represents that proposed system is more improved than existing system.

VI. CONCLUSION AND FUTURE WORK

In this paper an OTP based authentication model for enhancing the security of the cloud system has been proposed. Traditional OTP uses some form of encryption decryption approach. The proposed system handles OTP in different manner. In this system a key is provided to user at the time of registration that will be utilized to decode numbered OTP and entered by a virtual keyboard at the time of OTP authentication. Key will be utilized to decode numbered OTP into the actual. Numbered OTP will not require any encryption to store in the session; this is the main benefit of the system. The proposed system is speedy as well as more secure. Proposed work is evaluated as compared with traditional approach of OTP generation. Main limitation of the proposed method is to remember the key used for decoding the OTP.

The proposed system can offer a real added value is in authentication to cloud services. Businesses are increasingly moving critical applications to the cloud. On the one hand this saves cost; on the other hand this also means that authentication becomes critical to keeping confidential data safe. Again, in many modern office setups infrastructure may be shared, and workers are increasingly mobile (and use mobile devices), making it hard to enforce access the traditional way by granting access based on the IP address of the client. The proposed authentication concept can be enhanced with fake web page and tracing events done by user.

REFERENCES

[1] Keiko Hashizume, David G. Rosado, Eduardo Fernandez-Medina, and Eduardo B. Fernandez, "An analysis of security issues for cloud computing", Journal of Internet Services and Applications Vol 4, Issue 1, pp 1-13, 2013.

[2] Cheng-Yuan Ku, Yu-Siang Chiu, "A Novel Infrastructure for Data Sanitization in Cloud Computing," In Diversity, Technology, and Innovation for Operational Competitiveness: Proceedings of the 2013 International Conference on Technology Innovation and Industrial Management, pp. S3_25-28, 2013.

[3] Baker, M. Mackay, and M. Randles, "Eternal Cloud Computation Application Development." Developments in E-systems Engineering (DeSE), pp. 392-397, 2011.

[4] Verma and S. Kaushal, "Cloud Computing Security Issues and Challenges: A Survey", Proceedings of Advances in Computing and Communications, Vol. 193, pp. 44-54, 2011.

[5] CLOUD SECURITY ALLIANCE The Treacherous 12 - Cloud Computing Top Threats in 2016, © 2016, Cloud Security Alliance.
[6] Lt. Col. Jatinder Paul Singh, Dr. Mamta and Sunil Kumar3 "Authentication and Encryption in Cloud Computing", IEEE 2015.
[7] Chun-I Fan, Pei-HsiuHo, and Ruei-Hau Hsu, "Provably Secure Nested One-Time Secret Mechanisms for Fast Mutual Authentication and Key Exchange in Mobile Communications", IEEE/ACM Transactions on Networking, Vol. 18, No. 3, JUNE 2010.
[8] Prof. More V.N, "Authentication and Authorization Models", International Journal of Computer Science and Security (IJCSS), Volume 5, Issue 1, 2011.
[9] Lexus Jun Hong Sim, Shu Qin Ren, Sye Loong Keoh and Khin Mi ,Mi Aung, "A Cloud Authentication Protocol using One-Time Pad", IEEE-2016.