Indian Journal of Science and Technology, Vol 9(23), DOI: 10.17485/ijst/2016/v9i23/88322, June 2016
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Preventing Cloud Attacks using Bio-Metric Authentication in Cloud Computing

S. Srinivasan[1*] and K. Raja[2]

[1]Research and Development Center, Bharathiar University, Coimbatore - 641046, Tamil Nadu, India and Department of M. C. A., K. C. G. College of Technology, KCG Nagar, Rajiv Gandhi Salai, Karapakkam, Chennai - 600097, Tamil Nadu, India; effectivemail@yahoo.com
[2]Alpha College of Engineering, Udayavar Koil Street, Chennai - 602107, Tamil Nadu, India; raja_koth@yahoo.co.in

Abstract

Objectives: To provide a secured and efficient solution for end users to access their personal files in the cloud servers using biometric authentication. **Methods:** In this Minutiae Map (MM) algorithm is implemented for processing fingerprint based authentication. The user personal files are stored in free public multiple cloud storages namely Dropbox and CloudMe using splitting and merging techniques. RC4 algorithm is used to improve the security in cloud environment. Cross site request forgery (CSRF) and Cross site scripting (XSS) prevention techniques are used to provide security against cloud attacks. **Findings:** This study analyses that MM algorithm is the best accurate fingerprint feature extraction algorithm compared to Orientation Map, Gabor Filter and core point detection techniques. The proposed approach measures the user personal files upload time in cloud servers namely Dropbox and CloudMe. The study also analyses the presence of CSRF and XSS attacks in the application. **Applications/Improvements:** The proposed system can be improvised involving preventive measures for more security threats and integrating other biometric authentications.

Keywords: Cloud Computing, CloudMe, CSRF, Dropbox, Fingerprint, Minutiae Map, XSS

1. Introduction

Cloud computing plays an important role in daily life. Also cloud computing invokes various security threats/ vulnerabilities such as Phishing, Session Hijacking, Click Jacking, Malicious software (Malware) and wireless connection vulnerabilities. In this paper, we examine a secured architecture for storing end user essential data like Birth certificate, Death certificate, Community certificate, Academic certificate, Transfer certificate, Passport and other personal details in Cloud. The proposed architecture focuses on fingerprint based user authentication to access the data in the cloud. Now a day's fingerprint sensor has been integrated in all Laptops and mobile devices, so using the inbuilt sensor or integrating an external fingerprint sensor user can authenticate, store/ download the respective data.

The fingerprint feature extraction and matching is performed using Minutiae Map algorithm (MM). Minutiae is the reference to bifurcation and termination values of the ridges in the fingerprint. For each individuals a unique signature/pattern can be obtained from the distribution of their fingerprints. The distribution on the fingerprint provides a unique signature for each and every individuals[2]. Our proposed architecture also involves multiple cloud storage. The files are split into fragments and stored in Cloud server A and Cloud server B. The fragments are encrypted using RC4 algorithm and stored in the cloud servers.

Finally our proposed architecture provides a complete secure web solution invoking detection/prevention techniques for CSRF and XSS.

In[3] examined mobile based cloud computing. Fingerprint based authentication on mobile devices was

*Author for correspondence

implemented. In this research article, core point detection algorithm was used for fingerprint feature extraction.

In[4] presented a comparative study among the fingerprint feature extraction algorithms like gabor features, orientation maps, orientation collinearity and minutiae maps. The results showcase that orientation maps has least processing time among the fingerprint feature extraction algorithms.

In[5] proposed an architecture for storing the data in multiple distinct clouds simultaneously.

The idea of implementing multiple clouds was proposed however the previous works did not focus on security[6,20].

In[6] examined Identity-Based Distributed Provable Data Possession in Multicloud Storage. In few scenarios, the data owners stores their data in multiple cloud servers. Hence security, processing time and integrity checking protocol must be focused.

In[7], CSRF is defined as a malicious website causing a user web browser to perform an unwanted action. These attacks are also termed as "sleeping giant" because many websites on the internet cannot protect themselves since the prevention measures are totally ignored during web development and security communities. This paper recommended server-side changes that can be able to prevent the websites from CSRF attacks.

In [8], stated RESTful APIs are more widely implemented by the web developers to enhance functionality of the websites. XSS attacks take advantage of more characteristics of RESTful APIs. This research article deeply analysis the XSS attacks in RESTful. APIs. This paper proposed a tool to detect the XSS vulnerabilities in APIs automatically.

In [9], focused on cloud attacks, data integrity, data leakage, privacy, confidentiality, vulnerabilities during sharing of resources, services and information. The research paper assures security on transmission of data, quality of service, prevents vital information from various active and passive attacks.

The rest of the paper is organized as follows: Section 2 reviews the literature survey. Section 3 elaborates our proposed architecture. Section 4 reviews the fingerprint feature extraction, Cloud Storage. Section 5 examines the CSRF and XSS attacks and prevention techniques. Section 6 showcase the experimental results. Section 6 examines the conclusion.

2. Proposed Architecture

Our proposed architecture is described in the Figure 1.

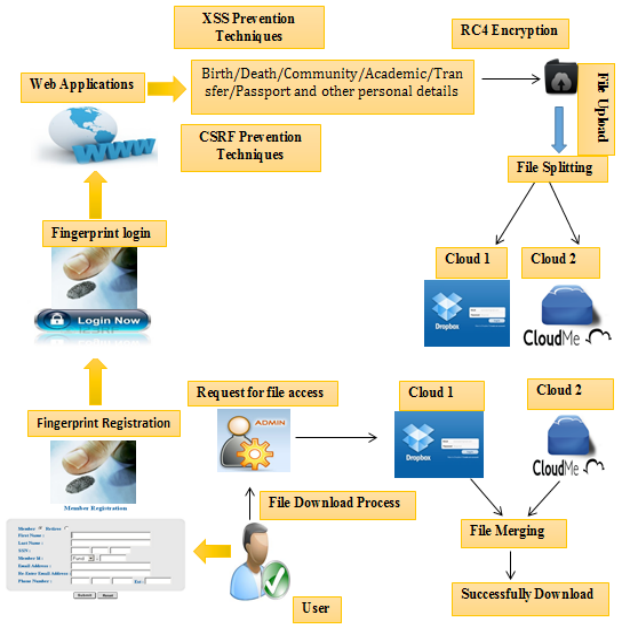


Figure 1. Proposed Architecture.

3. Fingerprint Feature Extraction

In our proposed system, the user fingerprint feature are extracted and verified using Minutiae Map algorithm (MM)[1,10,11]. Minutiae Map algorithm identifies the fingerprint ridges and extracts the bifurcation and termination values which is shown in Figure 2 from the input fingerprint image. Ridge termination is defined as the fingerprint point at which the ridge gets terminated or ends. Bifurcation is defined as the fingerprint point at which the fingerprint ridge gets split into two halves which is shown in below figure 2. Our proposed system provides the respective user fingerprint total bifurcation, termination values along with its location (X, Y coordinates) and stores in the database during user registration.

In our proposed system, ideal thinned ridge is considered. We assume usually a thinned ridge will have a value 1 or 0. The algorithm used 3*3 windows to scan the fingerprint image. In the output image, both termination and bifurcation points is denoted by a dot stating their exact appearance in the fingerprint image.

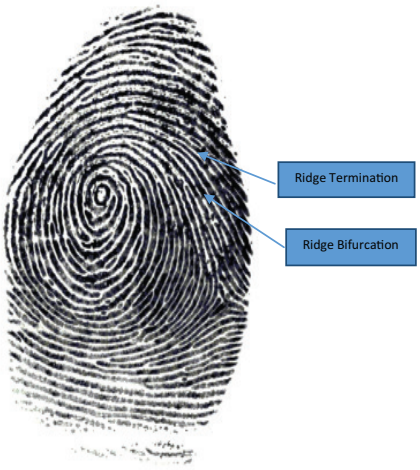


Figure 2. Fingerprint Bifurcation and Termination points.

The algorithm uses 3*3 windows to scan the image and the bifurcation and termination in the final output image shall be represented by a dot. Let's consider (x,y) denote the pixel on the thinned ridge and N_0, n_1, ……N_7 denote its neighbours.

A pixel (x,y) is a ridge ending if,

$$\sum_{i=0}^{7} Ni = 1$$

A pixel (x,y) is a ridge bifurcation if,

$$\sum_{i=0}^{7} Ni > 2$$

The fingerprint features are been extracted using MM algorithm and checked with the registered database for user identification. The matching score is been determined[12]. If the matching score is higher than the threshold value, the end user will be promoted to store their data in the cloud storage,

4. Data Storage

A cloud server is defined as an entity which contains huge data storage managed by cloud service provider. This paper focuses on providing high security for user data in public cloud storages because there is a huge demand of security prospects in public cloud storages[2]. Describes the security threats with regard to cloud computing.

Cloud Storage usually contains business-critical data and processes, hence high security is the only solution to retain strong relationship between the cloud service providers and data owners. Thus to overcome the security threats, this paper proposes **multiple cloud storage**[4]. In this all files and databases of a specific user is split and stored in various cloud storages (e.g. Cloud A and Cloud B).

Databases consists of tables, rows and columns. Databases are easy to store in multiple cloud storages. Our application will act as a combiner and store different parts of the table such as rows and columns in multiple clouds using **Vertical fragmentation** and **horizontal fragmentation**[6]. These rows and columns will be encrypted using **RC4** (Stream Cipher) encryption algorithm. During response our application combines the data and sends to the verifier.

Files are stored in multiple clouds using cryptographic data splitting. The file is split into fragments and stored in distinct cloud servers with encrypted key. Thus once the authorized token for the specific file is requested, the cloud server performs a keyword based search on the encrypted data's and combines the fragments. This is sent to the verifier.

5. RC4 Algorithm

Many cloud service providers (CSP's) are untrusted and privacy of the data owner information must be protected using encryption techniques[13]. RC4 is a state cipher which is also termed as stream cipher. RC4 was designed by Ron Rivest. In this plain text digits are encrypted and processed one at a time. The processing speed is higher than block ciphers. RC4 algorithm is well suited for real time processing. In RC4 both encryption and decryption is performed using the same algorithm.

The RC4 algorithm is of two stages:

- Initialization –Data encryption using a respective key, Creating arrays, Assigning values and selected key to the arrays.
- Operation – Swapping and XOR the final output to obtain the cipher text.

6. Security Analysis: CSRF and XSS

Providing cloud data security is very important. Thus identifying various security issues in cloud service delivery model is also important[15].

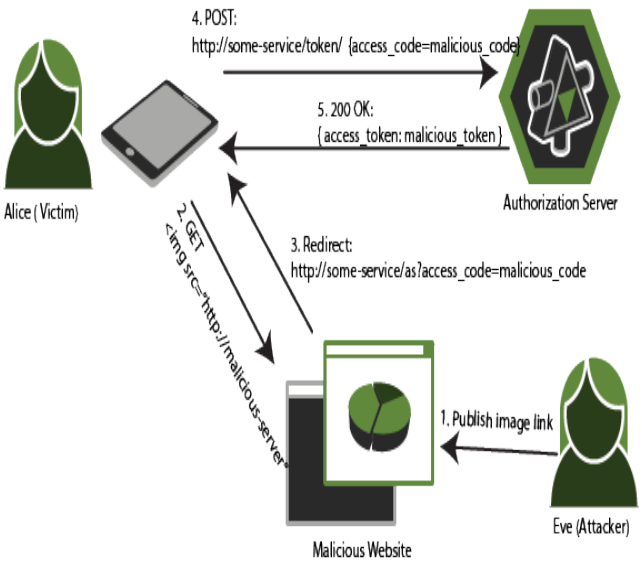


Figure 3. Process diagram for CSRF attack.

6.1. Cross-Site Request Forgery

CSRF is also denoted by one click attack or session riding. This attacking site sends request to the targeted website through web browser. This attack uses the authenticated session established between the web browser and website. This attack targets only on the functionality that causes a state change on the server such as changing the email address (or) password, or purchasing something etc..,. This attack specially target state change of the packet or the request not a theft of a data. Simply this attack can be demarcated as a packet that is captured and changes its values. The CSRF process is explained in Figure 3.

6.1.1. Prevention

There are various methods discussed here to prevent the CSRF attacks. Primarily, this attack can be prevented by using a secret cookie. A secure cookie can only be transmitted over an encrypted connections. By using URL writing (An incomplete solution since some session information included in the URL) one can able to avoid CSRF. CSRF Token protection bypass method must be used. But mostly this technique won't work. Finally by accepting the post request method one can avoid CSRF where the tokens are used to a large cryptographic value that are difficult to guess. Verifying the referral header, Secret Validation of tokens, Custom HTTP header, Http-Only Request are some of the effective preventing techniques for CSRF attack. Hence even if user access a link without his knowledge that contains CSRF flaw, then the web browser would not showcase the cookie to an external party.

6.2 Cross Site Scripting (XSS)

Nowadays hackers across globe use their skills/techniques/tools to attack the web portals/websites and web applications to pull the information without the user's knowledge. Web applications produces different outputs to the users according to some sorts of preferences and specific needs. Cross-site Scripting mainly attack the dynamic weaker websites and get serious vulnerabilities interpreting the organizations. Thus the XSS attack converts the web applications/websites into a victim. Thus XSS attack is defined as injecting malicious scripts into the targeted machine to steal cookies and collect the required/important data. The overview of XSS is shown in Figure 4.

6.2.1. Types of XSS

The below mentioned are the three different types of cross site scripting,

- Reflected
- Stored
- DOM Based XSS.

REFLECTED (Non persistence): In this type of XSS a pop up error message will blink and disappear. In this type the error message will be blinks as

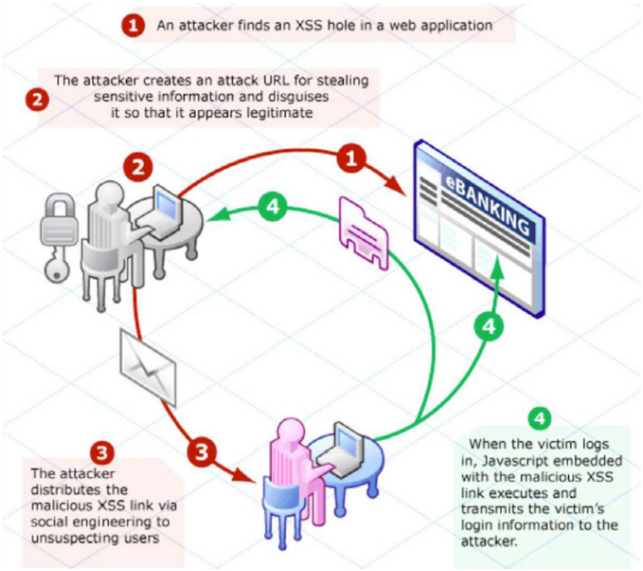


Figure 4. Process diagram for XSS attack.

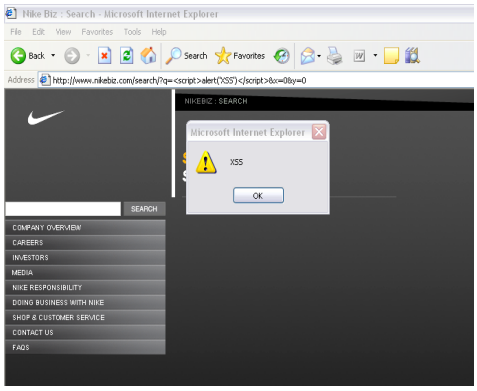


Figure 5. Reflected cross site scripting.

Exploit URL:
http://www.nikebiz.com/
search/?q=<script>alert('XSS')</script>&x=0&y=0

Figure 5 describes the reflected cross site scripting.

STORED (Persistence): In this type, a pop up message has stored in our inbox once a user login and the errorb-links overtime.The JavaScript supplied by the attacker is stored in the each website (e.g. in a database). It doesn't require the victim to supply the JavaScript somehow, it will just visit the exploited web pages which is moredangerous than Reflected XSS. As the result many XSS worms on high profile sites like Myspace and Twitter will occurs and affects these webpages.

DOM Based XSS: In this method, there is the possibility of content processing stages performed by the client. This type is not that much effective than Reflected and Stored.

6.2.2. Prevention Technique

Typical HTTP Request: The input validation by each user should be checked by manual and automated testing techniques.

Cookie Options Mitigate Input: Securing cookie is an effective method to prevent XSS attack.

Web Application Firewalls: By using web application firewalls the user can protect the old application. No General solution has found for XSS because one can't able to protect all the applications from the XSS attacks.

7. Experimental Results

Figure 6 shows user registration form. This allows the user to register into our application.

The registered users can login using the below Figure 7.

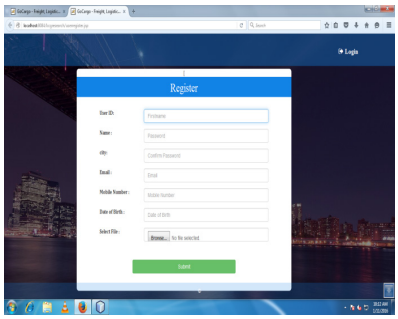


Figure 6. User Registration Form.

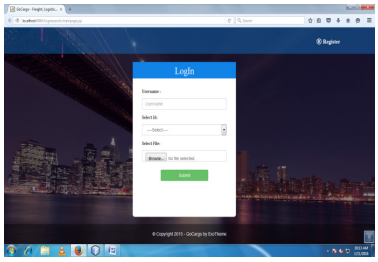


Figure 7. User Login Form.

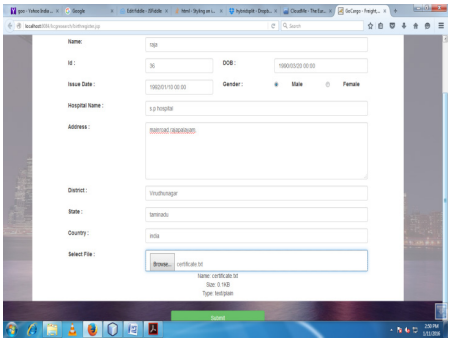


Figure 8. User personal details registration and upload form.

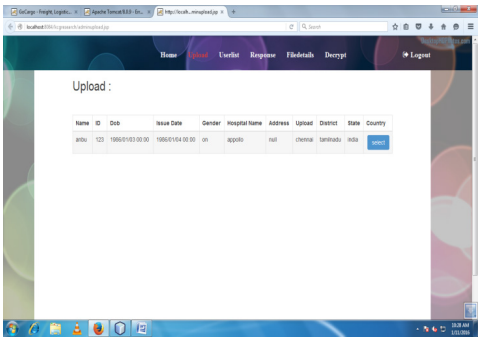
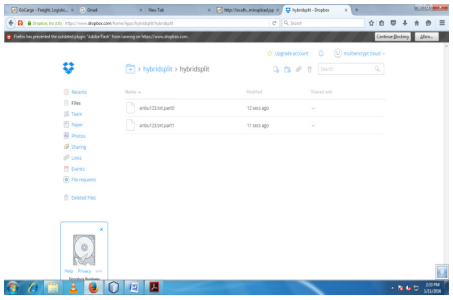


Figure 9. Uploaded user details.

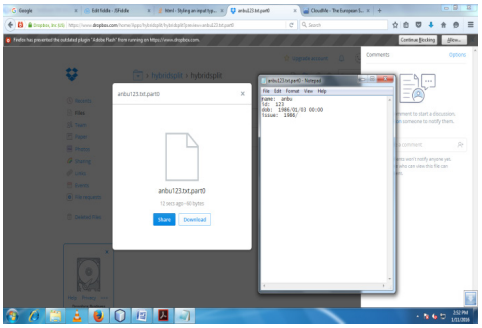
Figure 8 shows the form where user can provide the input file to be uploaded.

Figure 9 shows the uploaded details to the user for conformation before uploading into the cloud server.

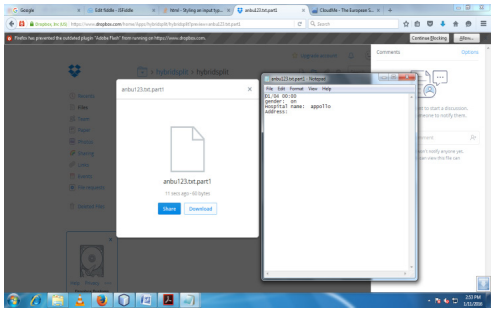
Figure 10 and 11 shows the user input files is been split into 2 parts and stored in dropbbox and cloudMe servers.



(a)

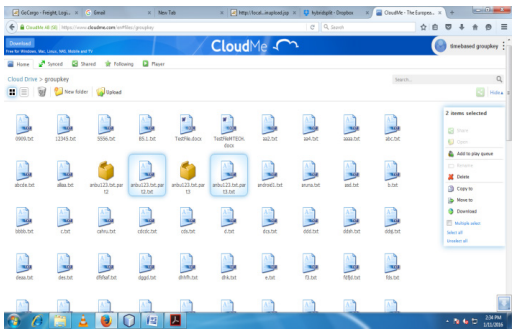


(b)

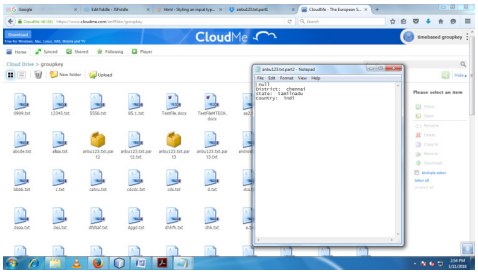


(c)

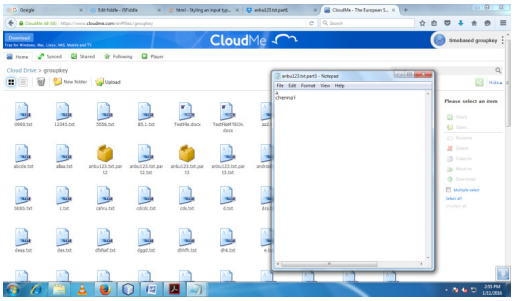
Figure 10. (a) The user details are split and 2 parts are stored in real public cloud named Dropbox. (b) & (c) The user details which has been split and stored in Dropbox cloud.



(a)



(b)



(c)

Figure 11. (a) The user details are split and 2 parts are stored in real public cloud named CloudMe (b) & (c) The user details which has been split and stored in CloudMe cloud.

Table 1. File upload time in CloudMe and Dropbox cloud servers

File Size	Upload time in Dropbox (ms)	Upload time in CloudMe (ms)
1 KB	3499	4296
3 KB	3738	4386
5 KB	3903	4454

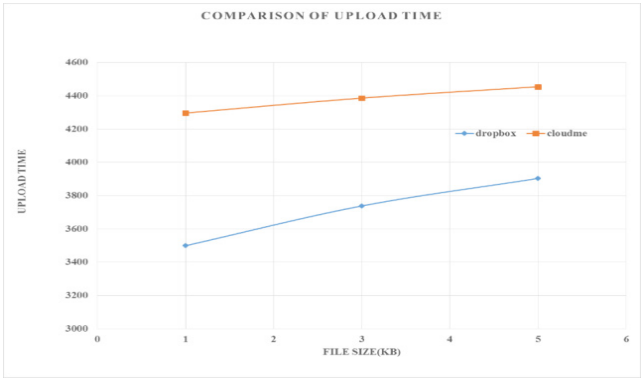


Figure 12. Representation of upload time in CloudMe and Dropbox cloud servers.

The Table 1 demonstrates the upload time of file size 1KB, 3KB, 5KB in dropbox and cloudMe servers.

Figure 12 demonstrates the upload time of different file types in CloudMe and Dropbox cloud servers.

7.1 CSRF

The CSRF attack has been implemented in the Login Page of the user since this attack targets the function of the state change for example changing the user password. Thus the experimental result has been processed in changing the password of the user by the attacker using CSRF are elaborated in below Figure 13, 14, 15 and 16.

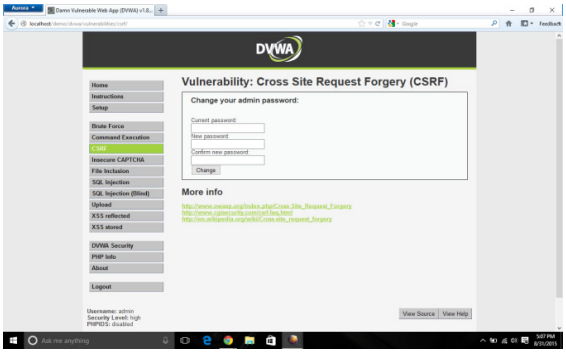


Figure 13. The user login to the browser to change his/her password. The user changing the new password as "HAI".

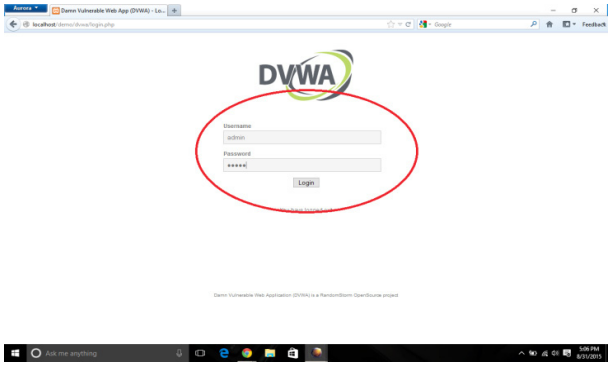


Figure 14. The password has been changed succesfully for the user.

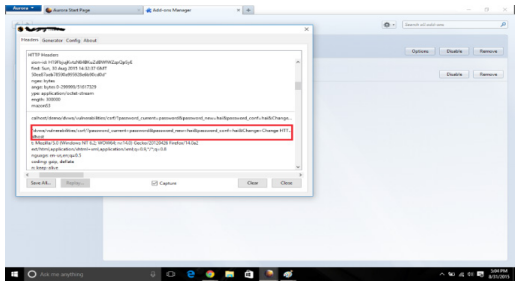


Figure 15. This image shows that how a request has been sent to the server in the backend process.

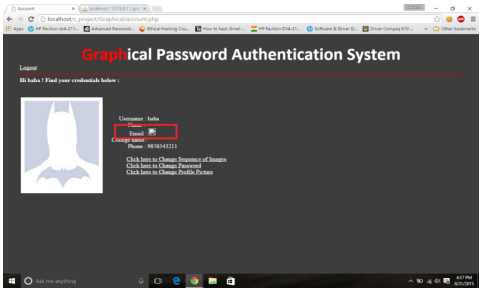


Figure 16. Here the attackers grab the http requested packet and reset the password. Thus the password "hai" is changed as "password". And the attackers sends the packect again to the server.

7.2 XSS

The experimental research of Cross Site scripting (xss) is a hacking process in which the attack is injected in the web page before the user login to that web site. This technique is mainly concentrate on change of script at the client side process. Figure 17, 18 and 19 describes the possibility of XSS attack in user login page.

Our literature survey states XSS and CSRF attacks are most common in web applications containing vulnerability. The Table 2 defines the percentage of vulnerability among web applicationsaccording to Trustwave Global Security Report[16].

As discussed in section 6.2.1. The cross site scripting types namely Reflected XSS and Stored XSS are been analysed in a live website named www.ieees.online and Table 3 shows the analyzed results.

Reflected XSS: In the website www.ieees.online, the vulnerable script is injected and the script reflects the

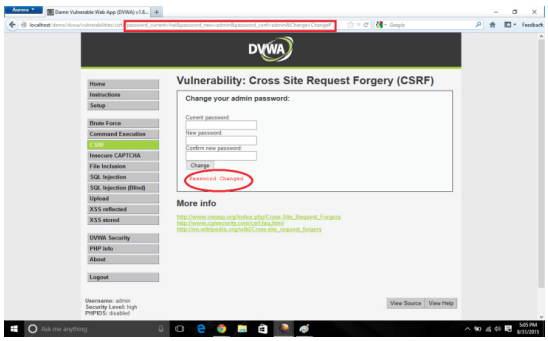


Figure 17. Script is been injected in the fields like Name, college name and e-mail.

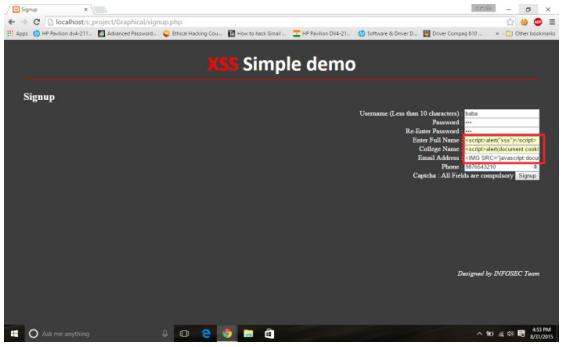


Figure 18. The script injected has been executed showing a pop-up message indicating the XSS attack has been occurred.

Figure 19. Due to the script injected the above figure shows that instead of e-mail address an image has been loaded.

Table 2. Percentage of attacks found in global applciations

Attacks Found	Percentage
SQL Injection	15%
Insecure Redirects (Phishing)	24%
Cross-Site Request Forgery	72%
Cross-Site Sripting (XSS)	82%
Other Injection	7%

Table 3. The results of vulnerable scripts for performing a XSS attack found in www.ieees.online website

Attack name	Domain	Vulnerable scripts	DIV	Span
Reflected Xss	www.ieees.online	21	40	16
Stored Xss	www.ieees.online	19	40	16

Table 4. The results after implementing the XSS prevention techniques

Attack name	Domain	Vulnerable scripts	DIV	Span
Reflected Xss	www.ieees.online	0	0	0
Stored Xss	www.ieees.online	0	0	0

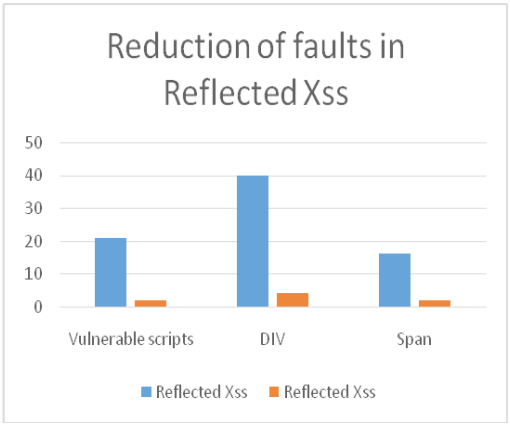


Figure 20. The reflected XSS script foundbefore and after implementing prevention techniques in www.ieees.online.

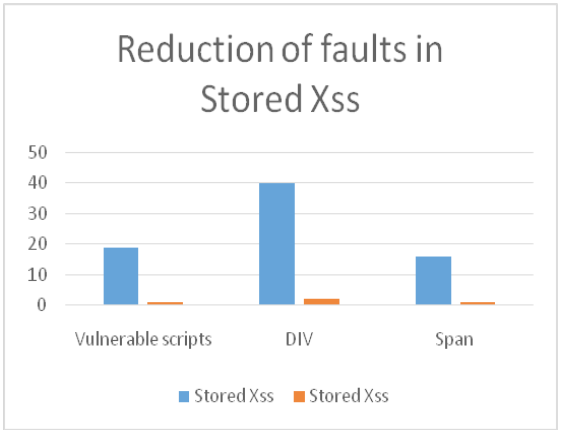


Figure 21. The Stored XSS script found before and after implementing prevention techniques in www.ieees.online.

particluar action assigned. After applying the prevention techniques the vulnerable scripts were blocked.

Stored XSS: In the website www.ieees.online, the vulnerable script was able to get injected and stored in database. After applying the prevention techniques, the special characters were removed and only values are stored in the database. The final results after invokiing

XSS prevention techniques in www.ieees.online is shown in Table 4.

Figure 20, 21 is an graphical representation of XSS vulnerabe scripts in www.ieees.online before and after applying the prevention techniques.

8. Acknowledgment

My sincere thanks to Stigmata Techno Solutions LLP to complete this paper.

9. Conclusion

This paper provides a unique solution invoking security in multiple cloud storages. Also our experimental results showcase Minutiae Map algorithm as secured and has less processing time for fingerprint feature extraction when compared with Orientation Maps, Gabor features and orientation collinearity algorithms. Our proposed architecture also involves multiple cloud storage system. The user files are converted into fragments, split and stored in 2 cloud storages namely Dropbox and CloudMe. Our experimental results show that storage time in Dropbox is less when compared to CloudMe. The user personal details are stored as cipher text in cloud servers using RC4 encryption algorithm. Our paper also deals enhancement of security to cloud servers by implementing detection and prevention techniques against CSRF and XSS attacks. Hence our architecture provides complete secure access/storage of user records in multiple clouds. In future, this proposed work can be extended for mobile based transaction.

10. References

1. Mell P, Grance T. The NIST Definition of Cloud Computing. Version 15, Nat'l Inst. Of Standards and Technology, Information Technology Laboratory. 2010; 53:50. Available from: http://csrc.nist.gov/groups/SNS/cloud-computing/.
2. Hubbard D, Sutton M. Top Threats to Cloud Computing V1.0. Cloud Security Alliance. 2010; p. 1-14. Available from: http://www.cloudsecurityalliance.org/topthreats.
3. Chi PW, Lei CL. Audit-Free Cloud Storage via Deniable Attribute-based Encryption. IEEE. 2015 Nov; 2(11):1-14.
4. Bernstein D, Ludvigson E, Sankar K, Diamond S, Morrow M. Blueprint for the Intercloud-Protocols and Formats for Cloud Computing Interoperability. Proc. Int'l Conf. Internet and Web Applications and Services. Venice. 2009; p. 328-36.
5. Bohli JM, Gruschka N, Jensen M, Iacono LL, Marnau N. Security and Privacy-Enhancing Multicloud Architectures. 2013 Aug-July; 10(4):212-24.
6. Wang H. Identity-Based Distributed Provable Data Possession in Multicloud Storage. IEEE. 2015; 8(2):328–40.
7. Zeller W, Felten EW. Cross-Site Request Forgeries: Exploitation and Prevention. 2008; p. 1-13. http://citp.princeton.edu/csrf.
8. Zhang Y, Luo Q, Liu Q, Wang X. Cross-Site Scripting Attacks in Social Network APIs. 201 Sep.
9. Srinivasan S, Raja K. An Advanced Dynamic Authentic Security Method for Cloud Computing. New Delhi: Proc. 50th Golden Jubilee Annual Convention – Theme – Digital Life. 2015 Dec.
10. Yager N, Amin A. Fingerprint verification based on minutiae features: a review. Pattern Anal Appl. 7:94–113. 2004 Feb 14; 7(1):94-113.
11. Rassan IA, Shaher HA. Securing Mobile Cloud Using Finger Print Authentication. International Journal of Network Security & Its Applications (IJNSA). 2013 Nov; 5(6):41-53.
12. Rajanna U, Erol A, Bebis G. Springer: A comparative study on feature extraction for fingerprint classification and performance improvements using rank-level fusion. 2009 April 28.
13. Manjusha R, Ramachandran R. Secure Authentication and Access System for Cloud Computing Auditing Services Using Associated Digital Certificate. Indian Journal of Science and Technology. 2015 Apr; 8(S7). DOI: 10.17485/ijst/2015/v8iS7/71223.
14. Stallings W. Upper Saddle River, New Jersey, Prentice Hall: Cryptography and network security: Principles and practice. 2003.
15. Durairaj M, Manimaran A. A Study on Security Issues in Cloud Based E-Learning. Indian Journal of Science and Technology. 2015 Apr; 8(8). DOI: 10.17485/ijst/2015/v8i8/69307.
16. Barnett R. Black Hat, USA: The Web IS Vulnerable: XSS on the Battlefront (Part 1). 2013. Available from: https://www.trustwave.com/Resources/SpiderLabs-Blog/The-Web-IS-Vulnerable--XSS-on-the-Battlefront-(Part-1).