ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

Secure Data Storage Framework using Anti-XSS in cloud

P.Risha Hebiya[1], J.Ganesh[2]

P.G. Student, Dept. of Computer Science and Engineering, Arasu Engineering College, Kumbakonam, India[1]

Assistant Professor, Dept. of Computer Science and Engineering, Arasu Engineering College, Kumbakonam, India[2]

ABSTRACT: Cloud computing is an important model for delivering valuable business services to the cloud users. Cloud service consider as online Storage where the data is remotely stored, managed and backed up based on pay per use .The user to store their files in online and access them from anywhere via the internet. Data Storage Security is a primary concern when entrusting an organization to keep track their sensitive information in cloud but not under the direct control of that organization.The main challenge of cloud to ensure the correctness of user's data in the cloud storage. It focuses on cloud data storage security, always which has been an important aspect of quality of service. Data owners to store their valuable data in the cloud, then the problem arises how to protect those valuable data against security attacks. Cross site scripting (XSS) attack is a very dangerous vulnerability found in Web Applications, 'XSS' permits the attacker to inject malicious code into the client browser and execute that code. For that, to prevent cross site scripting attacks (Persistent and Non-Persistent attacks) by using Anti-cross site scripting mechanism. The proposed "Secure Keyless Algorithm" represents a new way of data protection in cloud storage. In this case, the data itself to create a protective shield to ensure the data integrity in cloud.The role of cloud audit server performs the user authentication and monitors the stored data in remote server. The proposed mechanism to be enhanced the data storage security in cloud. This paper shows results of proposed mechanism gives more security compared to previous techniques effectively.

KEYWORDS: Cloud Data storage, Cross site Scripting (XSS), Auditing.

I. INTRODUCTION

Cloud computing used in advanced technologies like virtualization, storage, connectivity and processing power are combining to create a new technical Ecosystem.Cloud computing is not a new technology, but a relatively new way of delivering services to the organization. Storage security is a primary concern while increases of security attack in the online world. The cloud offers larger amounts of data storage for users and organizations to store their sensitive information in cloud storage. This centralization of storage infrastructure results in cost efficiencies in utilities, wherever to access those stored data.[1],[4],XSS is "Cross-site scripting" (XSS) is a type of computer security vulnerability typically found in Web applications (such as web browsers through breaches of browser security) that enables attackers to inject client-side script into Web pages viewed by other users. Cross site scripting (also referred to as XSS) vulnerabilities happen when attacker uses web application to send malicious code in the form of script or special characters that would expose the internals of web site. Cross Site Scripting (CSS for short, but sometimes abbreviated as XSS) is one of the most common application level attacks that hackers use to sneak into web applications today. [2],[4],[12]Cross site scripting is an attack on the privacy of clients of a particular web site which can lead to a total breach of security when customer details are stolen or manipulated. The goal of the CSS attack is to steal the client cookies, or any other sensitive information, which can identify the client with the web site.Cross-Site Scripting (XSS) vulnerabilities are being exploited by the attackers to steal web browser's resources such as cookies, credentials etc. by injecting the malicious script code on the victim's web applications.

II.RELATED WORK

Adam Kie˙zun,Philip J. Guo[2], proposed a technique for finding security vulnerabilities in Web applications. To consider SQL Injection (SQLI) and cross site scripting (XSS) attacks are widespread forms of attack in which the attacker crafts the input to the application to access or modify user data and execute malicious code. To implemented a technique for PHP, in a tool Ardilla evaluated the web vulnerabilities. Ardilla's taint propagation tracks the flow of

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

tainted data through the database. The user of Ardilla needs to specify the type of attack (SQLI, first order XSS, or second-order XSS), the PHP program to analyze and the initial database state. Tejinder Singh Mehta , Sanjay Jamwal[3],They focused the possibilities of securing web applications on client side as well as on server side. To minimize theft space for taint contents by using QualysGuard (WAS) tool.

In cloud computing scenario QualysGaurd WAS acts as a software as-a-service (SAAS). QualysGuard WAS automates the websites scanning process QualysGuard WAS tool is the best tool to check browser with three options, Basic Scan:(browser issues). Intermediate Scan(browser issues and system settings)Advanced Scan (browser issues and system settings to reflect the issues with fix the issue). Bhanu Prakash Gopularamand N. Nalini [5] provide an approach to address cross site scripting security vulnerabilities security risks by Enterprise Security API from Open Web Application Security Project**.** Open Web Application Security Project (OWASP) is an open forum focused on security risks. The OWASP Enterprise Security API (ESAPI) is a free open source library designed to be used in web projects and mitigate the cross site scripting issues. Cong Wang, QianWang[6] ,to ensure the correctness of user's data in the cloud. To detect any unauthorized data modification and corruption in stored data. So, They provide efficient mechanisms for dynamic data verification operation to achieve their goals such as Storage correctness and Fast localization of data error. Jin Li, Xiao Tan [7] describe the integrity of data storage in cloud computing.To reduce the computational cost at user side during the integrity verification of their data.The proposed OPoR scheme is proved security against reset attacks and reduces computation overhead for tag generation on the client side.Neha, Paramjeet Singh [8] to concentrate on data protection mechanism and proposed new optimal keyless algorithm. The algorithm had been implemented over 128 bit blocks .They proposed with the motive to provide highest security level with minimum execution time in terms of encryption and decryption. Akhil Kaushik, Satvika [9], to introduced KUDOS cryptographic algorithm basically falls under the symmetric encryption, the same key is used at both ends to encrypt and decrypt the data. The proposed algorithm was Keyless User Defined Optimal Security Encryption (KUDOS) is based on the concept of user customization. The algorithm can be implemented with or without error control. C wang, Sherman S. M. Chow[10] describe the cloud data storage security with Privacy Preserving Public Auditing with watermarking for data Storage security in cloud computing. The watermarking processes achieve the very high security level so the data or images cannot be identified by the intruder in the cloud. It supports data dynamics where the user can perform various operations on data like insert, update and delete.Mohammad Reza Faghaniand UyenTrang Nguyen[11] explore the possibility of using the selective monitoring approach instead of the exhaustive checking method. In particular, they present a study of potential selective monitoring schemes. In a selective monitoring scheme, they do not monitor every read/write post or every user in the OSN(Online Social Network). Instead, monitor only a subset of users and their friends' activities.

II. PROBLEM STATEMENT

System Model:

To provide anti cross site scripting mechanism for storage server in cloud environment and to store the data using secure key-less(SKA) algorithm.

Client: An entity like individual or organization to store large amount files or data in the cloud storage and access those stored data wherever via internet.

Cloud audit server: To perform authentication for the user at the time of cloud login. If any unauthorized users try to access the cloud storage means the cloud audit server to reject that user by using anti cross site scripting mechanism. The authentication must be performed in a secure manner. In case the client does not necessarily monitor their data, they can hand over the monitoring task to a trusted cloud audit server.

Cloud Storage Server:

After the user authentication performed by the cloud audit server then user or organization to store the data into the cloud storage server and retrieve wherever user want to access those files.In cloud storage, by putting the large data files onthe remote servers and the user can be relieved from the burdenof computation and storage. As clients no longer possesstheir data locally, it is of critical importance for the clients to ensure that their data are being correctly stored and maintained.

Attacker:

An attacker who uses cross-site scripting successfully might compromise confidential information, manipulate or steal cookies, create requests to valid user and execute malicious code on the end-user systems.

Contributions:

Our contribution can be summarized as follows:

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

- ✓ In cloud auditor act as intermediate between the client and storage server.Which is responsible for the user authentication at the time of user login to the cloud and monitoring the cloud storage.
- ✓ To provide the security against cross site scripting attacks (Persistent and Non Persistent) in cloud storage server by using anti cross-site scripting mechanism.
- ✓ The proposed "Secure key-less algorithm" provide data itself to create a protective shield and to ensure the data integrity in cloud storage.

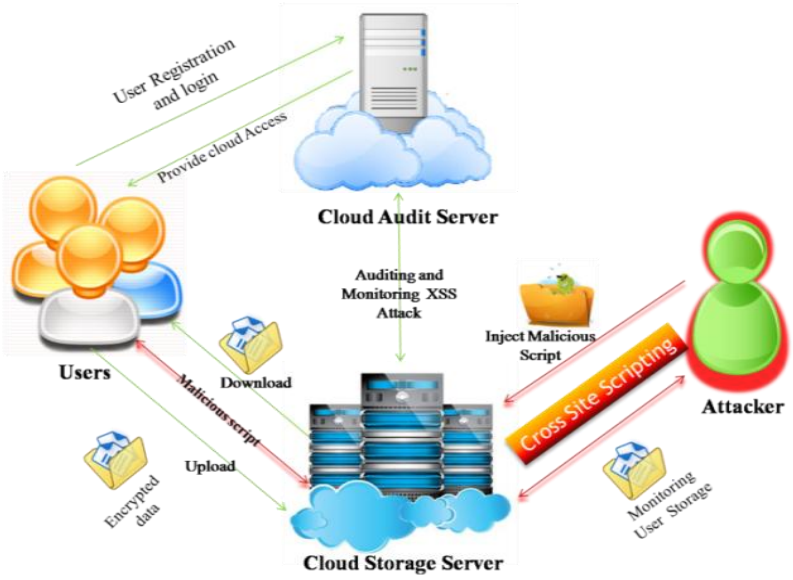


Fig.1.System Architecture

In Fig.1.shows the attacker to insert the malicious script to the cloud server.When the cloud user want to access the cloud storage at the time the malicious script to be downloaded from the storage.Then the downloaded script will be run onto the user system.After that the attacker collect the user credential data from that user system.In the cloud paradigm, by putting the large amount of data on the remote servers, so the user can be relieved from the burden of storage and their computation.As user no longer possess their data locally, it is of critical importance for the user to ensure that their data are being correctly stored and maintained by remote server. In case those users do not necessarily have the time or resources to monitor their stored data, they can hand over the monitoring task to a trusted cloud audit server of their respective choices.

In this paper we consider the secure data storage framework against cross site scripting attacks by using anti cross site scripting mechanism and to encrypt the data using secure key-less algorithm (SKA) for provide secure data storage in cloud environment.

CROSS SITE SCRIPTING (XSS):

Cross site scripting (XSS) permits the attacker to inject malicious code (Java Script) into the user system or targeted cloud storage server. The reason for that is the user trusts the remote storage server to be fully secured. Once the attacker to import the malicious code into the targeted private cloud server means it will be spread onto the entire organization employees who are all to access that server.Then the malicious code will be automatically running under the browser until it was removed.The client browser execute the malicious code and collect the user credential (data). V. Nithya, S. LakshmanaPandian[1], XSS can be defined as a security exploit in which an attacker can embed malicious script (JavaScript, VBScript, ActiveX, HTML, or Flash) into a vulnerable dynamic page.In our paper, we concentrate on two kind of cross site scripting attacks and their prevention mechanisms.NunoAntunes and Marco Vieira [13],To describe two most common risks in the Web environment, injection-namely SQL injection, which lets attackers alter SQL queries sent to a database and cross-site scripting (XSS), are also two of the most dangerous. XSS vulnerabilities exist when an application sends user-supplied data to a Web browser without first validating or encoding that content.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

1.Persistent XSS Attack:
In the persistent XSS an attacker can inject the malicious code into the page permanently and that harmful code will be stored in the target cloud storage servers as an html text, Javascript in a database .When the user to give the request to the target cloud storage server at the time the malicious content to move onto the corresponding user system.Then the malicious code will be run on the user victim. This code will be stored in the page which will show to the user victim later on. If the "STORED XSS" vulnerability is successfully exploited by attackers, it will persistently attack the users until administrator remove that vulnerability. The Fig.1 shows the functionality of cross site scripting attacks in cloud storage.

2.Non-persistent XSS Attack:
The attacker sends a link to the corresponding user (e.g., Email) by social network website.When the user to open that link the web page contain malicious content and some hyperlinks within that webpage. If the user clicks on the link or hyperlink, the vulnerable scripts or malicious code will be automatically downloaded to the user system by web browser and run behind it. Then the user original data and stored cookies information's are sent to the attacker. It is otherwise called as "REFLECTED XSS".

III. SECURITY MODEL

(1).Cross-site Scripting (XSS) Attack Detection:

In order to detect Cross-site scripting (XSS) attacks is a difficult one but the filtering of XSS vulnerabilities and other detection methods are being researched. This section explains the related detection technique.

i.Lattice-based Approach: There is a tool called WebSSARI which combines static and runtime features and find security vulnerabilities by applying static taint propagation analysis.WebSSARI follows type state and lattice model and uses flow sensitive, intra-procedural approach to determine vulnerability. When this tool knows that tainted data has reached sensitive function, it automatically puts runtime guards which are also called as sanitization routines [1].

(2).Cross-site Scripting (XSS) Attack Prevention:
There are main criteria that can be used for differentiate XSS prevention techniques is at point of deployment server-side and client-side and both.

i.Secure XSS-Tool:

We proposed secure XSS-Tool to find the cross site scripting attacks from client side.Then to ensure the security of user web browser.This tool to detect the cross site vulnerabilities (XSS) in web browser and intimate some, "Alert Message" to the corresponding user web browser.The secure XSS-Tool has set of taint data information's database based on that to check the web sites running on the web browsers. If any taint or malicious scripts found means to block that websites and create some warning messages to the user. That tool to be implemented for client side (Non-Persistent) XSS attack prevention.

ii.Anti-XSS Framework:

In our paper we proposed a secure Anti-XSS framework for avoid the cross site scripting attacks in cloud environment. That framework provides good security to the cloud storage server and detects the cross site scripting (Persistent attack) efficiently by cloud audit server.In this case the cloud audit server act as intermediate between the cloud storage server and cloud users. The role of cloud audit server to monitors the cloud storage server and prevents the cross site scripting attacks from hackers and provides the anti-cross site scripting mechanism to the cloud storage. If any attacker to inject the malicious code into the targeted cloud storage server means that proposed framework didn't allow that changes from others without proper authentication of cloud access by cloud auditor.

Secure Key-less Algorithm:

In our paper we proposed secure key-less algorithm for effective storage of data in cloud storage server. To consider the given Fig.2 represents the functionality of Secure Key-less algorithm (SKA) Encryption operations without any key and apply different kind of logical operation as well as shift functions to the data.

a)Encryption Algorithm:
To perform the following steps and to convert the original data into another encrypted data format.The Fig.2 shows the graphical representation of encryption algorithm from upload phase (Sender's Side).In our paper we take text data (256 Bit Block)as input and split into 120-16-120 sub blocks.

Step 1: The original data (Input) is arranged into column matrix.
(i.e.)characters are positioned in matrix form and columnar transposition is applied.
Step 2: The characters are converted into ASCII values.
Step 3: Then ASCII values are converted into binary values (0's and 1's).

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

Step 4: The binary data is splitting into 256 bit blocks. (The total number of bits divided by 256 bit blocks)
Step 5: Divide 256 bit block further into 120-16-120 sub blocks.
Step 6: For **odd round**

Apply 1's complement for the first 8bits of middle 16 bits

For **even round**

Get the binary code for gray code of second 8 bits of middle 16bits binary data.

Step 7: Perform XOR operation on middle first 8 bits and left 120 bit blocks.
Step 8: Then to perform left shift of the first 128bits and to perform right shift for second 128bit block.
Step 10: Last, each128bits are converted to gray code.

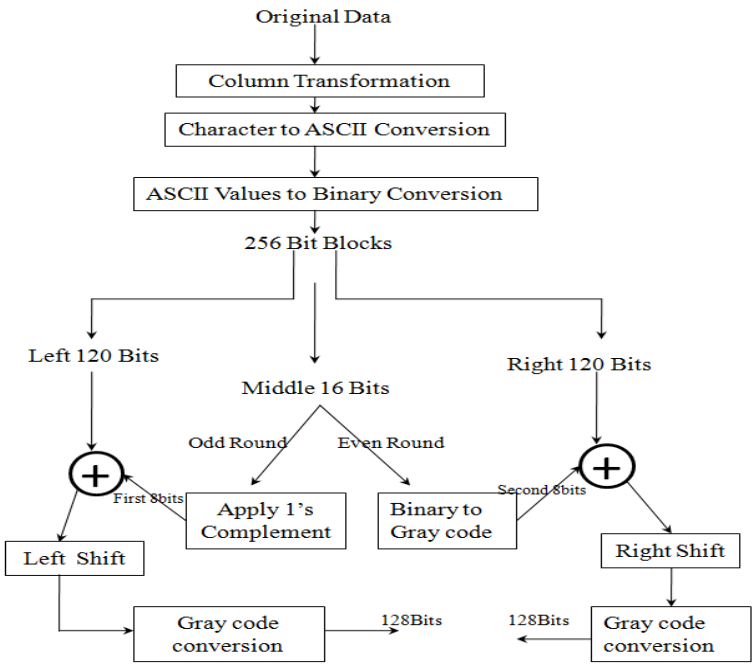


Fig.2.Encryption Function

b) Decryption Algorithm:

The decryption algorithm follows the same steps that are used in encryption algorithm but in reverse order. The Fig.3 represents the detailed functionality of decryption process without any usage of keys.In this case to consider the encrypted data (cipher text) as 256 bit blocks and to calculate the total number of rounds and then to apply different logical operation on that data. At last to decrypt the cipher text and get the original data (Receiver's Side).

Step 1: The total number of bits in the cipher text is divided by 256 to get the total number of rounds.

(i.e.)Encrypted Data (Cipher text) consider as 256 bit blocks.

Step 2: Divide the encrypted bit stream into 128 bit blocks.
Step 3: Then divide each 256 bit cipher text block into 120-16-120 bit sub blocks.
Step 4: For left 128 bits, gray to binary code conversion is performed and then right circular shift is done on this data. The shift size is equal to the number of rounds calculated in step 1.
Step 5: For last 128 bits, gray to binary code conversion is performed and then left circular shift is done on this data. The shift size is equal to the number of steps calculated in step 1.
Step 6: Leftmost 8 bits of sub block obtained in step-3 are performed XOR with the 120 bit sub block obtained in step-2.
Step 7: For **even round**,

Apply 1's complement for the first 8bits of middle 16 bits.

For **odd round,**

Get the gray code of the second 8bits of middle 16 bits.

Step 8: Repeat the steps 3-7 for number of rounds calculated by the algorithm.
Step 9: From the binary data, binary to ASCII and ASCII to Decimal conversion is done.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

Step 10: Character level decryption is performed to get the original data.
After the all the operations are performed then to get the original decrypted data (Download Phase) from step-10. In this kind of decryption process there is no need for the key generation and key transfer to decrypt the cipher text.Only to follow the some set of given procedures to change the cipher text to plain text.

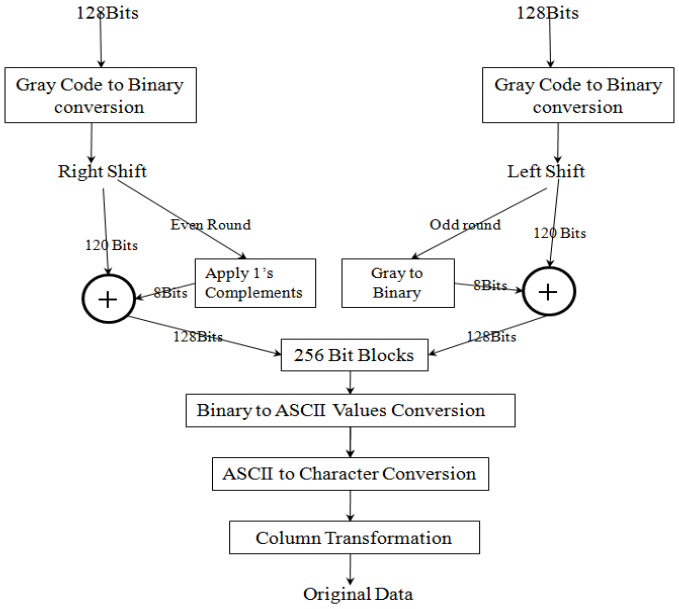


Fig.3.Decryption Function

IV. PERFORMANCE ANAYLSIS

To analyze the performance of proposed mechanism to enhance the security against cross site scripting attacks in cloud storage.Then consider the secure key-less algorithm(SKA) to provide the effective data storage in cloud.We compare our proposed algorithm gives better performance results with existing systems. The given graph clearly shows the improvement in encryption time and decryption time of the proposed algorithms based on the input data.We can see the time (Encryption and Decryption) difference between the existing system and proposed system. The table1 shows the different input data and their existing and proposed encryption times effectively.Compared to previous technique our proposed algorithm takes minimum time for encryption and decryption. Different text inputs of different sizes are taken then the results are calculated and compared by using two parameters for encryption time and decryption time have been shown below:

1.Encryption Time: The time consumed for the original data to converted into cipher data.
2.Decryption Time: The time consumption for convert cipher data back to the original data.

Input Size (in bytes)	Encryption Time (Existing System)	Encryption Time (Proposed System)
328	0.115416	0.045728
561	0.198228	0.076743
899	0.264142	0.168722
1535	0.6257507	0.470493
1873	0.5018402	0.450324

Table 1: Comparison between commonly used algorithm and proposed algorithm in terms of encryption time

Input Size (in bytes)	Decryption Time (Existing System)	Decryption Time (Proposed System)
328	0.112998	0.029471
561	0.158887	0.072092
899	0.218725	0.158875
1535	0.595787	0.451762
1873	0.549842	0.481288

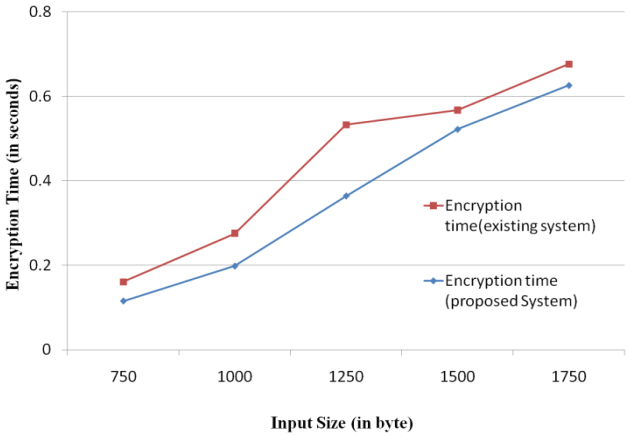
Table 2: Comparison between commonly used algorithm and proposed algorithm in terms of decryption time

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

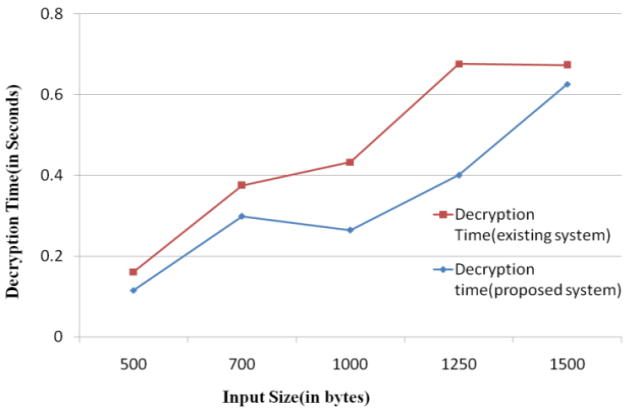
International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015



(a) Graphical representation of table 1



(b) Graphical representation of table 2

Fig.4

The above Fig.4 shows the results of improvement in encryption and decryption time parameter of the proposed algorithm as compared to commonly used optimal key-less algorithm for security. Different text inputs have been taken to analyze the performance of proposed system and existing system. The results are compared and analyzed on the basis of above mentioned metrics. The Fig.4(a) shows the graphical representation of encryption time based on given table 1 different input measurements. Similarly the Fig.4(b) shows the graphical representation of decryption time based on given table 2 different input measurements.

V. CONCULUSION

In this paper we analysis different kinds of cross site scripting attacks and their functionality. The proposed new framework to protect cloud data storage server against cross site scripting attacks (Persistent and Non-Persistent) and improve the storage security. The Secure XSS-Tool to detect the web vulnerabilities in clients side and Secure Anti-XSS framework to eliminate the cross site scripting vulnerabilities from cloud storage(Server Side) by cloud audit server. In our paper the cloud audit server performs the auditing operation and monitors the cloud storage server. Then clients relived from overhead the key generation and key management on behalf of the SKA(Secure key-less Algorithm).The algorithm provides security at both character level as well as bit level.For that to increase the security level is a major advantage of this algorithm. The results shown in the last section proves better performance of the proposed algorithm in terms of encryption time and decryption time.

REFERENCES

[1] Nithya.V, LakshmanaPandian.S and Malarvizhi.C "A Survey on Detection and Prevention of Cross-Site Scripting Attack ", International Journal of Security and Its Applications ,Vol.9, pp.139-152,2015.

[2] Adam Kie˙zun,Philip J. Guo,KarthickJayaraman,Michael D. Ernst," Automatic Creation of SQL Injection and Cross-Site Scripting Attacks",IEEE International Conference on Software Engineering,Vol.10,pp.199-209 August Vol.10,2010.

[3] Tejinder Singh Mehta , Sanjay Jamwal,"Model To Prevent Websites From XSS Vulnerabilities", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol.6, pp.1059-1067, 2015.

[4] Ahmed Elhady Mohamed," Complete Cross-site Scripting Walkthrough",website: www.infosec4all.tk, 2008.

[5] BhanuPrakashGopularam and Nalini.N," Cross Site Scripting Security Vulnerabilities and Risk Mitigation Using Enterprise Security API for Web-Apps on Public Infrastructure",International Conference On Emerging Research in Computing,InformationCommunication and Application(ERCICA),pp.790-794,2013.

[6] Cong Wang, Qian Wang, and KuiRen and Wenjing Lou," Ensuring Data Storage Security in Cloud Computing", IEEE Transactions onComputers,Vol.3,pp.1-9,Jul 2009.

[7] Jin Li, Xiao Tan, Xiaofeng Chen, Wong D.S, FatosXhafa "OPoR: Enabling Proof of Retrievability in Cloud Computing with Resource-Constrained Devices", IEEE Transactions on Cloud Computing, Vol.3, pp.195 – 205, April 2015.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

[8] Neha, ParamjeetSingh,Shaveta Rani, "Optimal Keyless Algorithm for Security",International Journal of Computer Applications,Vol.124,pp.28-32,August 2015.
[9] AkhilKaushik, Satvika, Manoj Barnela, and Anant Kumar," Keyless User Defined Optimal Security Encryption", International Journal of Computer and Electrical Engineering, Vol.4,pp.99-103,April 2012.
[10] Wang.C.,Chow.S.M.,Wang.Q.,Ren.k and Lou.W., "Privacy-Preserving Public Auditing for SecureCloud Storage",IEEETrasaction on Computers,Vol.62, pp.362-375,2013.
[11] Mohammad Reza Faghaniand UyenTrang Nguyen," A Study of XSS Worm Propagation and DetectionMechanisms in Online Social Networks", IEEE transactions on information forensics and security, Vol.8,pp.1815-1826, November 2013.
[12] David Gillman, Yin Lin, Bruce Maggs andSitaraman.k, "Protecting Websites from Attack with Secure Delivery Networks", IEEEcomputer society,pp.26-32,April 2015.
[13] NunoAntunes and Marco Vieira,"Defending against Web Application Vulnerabilities",IEEE Computer Society,pp.66-72,February 2012.
[14] Savage Rd. Ft. Meade,"Protect against Cross site scripting attacks",www.nsa.gov, sep-2010.