IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-5, Issue-4 E-ISSN: 2347-2693

Improving Source Code Encryption using Proposed Cipher Logic

SU. Rasal[1], R. Redhu[2], VS. Rasal[3], ST. Shelar[4]

[1*] Department of Computer Engineering, Bharati Vidyapeeth University College of Engineering, Pune, India
[2] Department of Computer Engineering, Bharati Vidyapeeth University College of Engineering, Pune, India
[3] Department of Computer science & Engineering, Nehru College of Engineering & Research Centre, Thrissur, India
[4] Department of Information technology, D Y Patil College of Engineering Akurdi, Pune, India

[*]Corresponding Author: surasal@bvucoep.edu.in Tel.: +918793000079

www.ijcseonline.org

Received: 19/Mar/2017, Revised: 26/mar/2017, Accepted: 19/Apr/2017, Published: 30/Apr/2017

***Abstract*—** Web page is loaded into web browser to run the required output according to Document Object Model. Web page contents source based on applied technology like language support, platform required and versions used. Indirectly, data can be anything but it is always in the form of 0 and 1. In this paper, proposed cipher logic is applied to the bits data directly due to which it become universal supportive. Five levels proposed cipher logic has been added to the encryption process. Same logic is required to decrypt the encrypted data without which original data cannot be retrieved.

***Keywords*—**Cipher policy, Document Object Model (DOM), source code, internet languages, web browser

I. INTRODUCTION

Web technologies vary according to requirement like php, jsp and others. In every document object model, source code is required to load into web browser. Web browser is an application to interact between user system and external network. Source code is focused to apply proposed cipher logic. Binary bits are considered to apply security logic. Same logic will be applied at both ends including sender and receiver to encrypt and decrypt respectively. Existing security techniques are used with applied cryptographic techniques like attribute based encryption with cipher policy [1]. Some cryptographic proposed approaches are explained like multiple attributes schemes with one time password approach [2].

II. SOURCE CODE EXECUTION NVIRONMENT

Current computing trend requires applied cryptographic techniques with emerging trends. Internet of things covers almost all computing streams like embedded systems, internet technologies, operating systems and others. It needs some security approach to prevent intrusions to the sensational things. Some applied cryptographic techniques are proposed like securing it with attribute based encryption and one time password approach and so on [3], [4].

A) Internet source code environment

Internet source code is a code used to develop a web page containing markup languages and style sheets. A web page is a document that is preferable for the World Wide Web which is stored on a web server and can be viewed using a web browser. Design of a web page requires a deep understanding of client requirements and the different platforms on which it can be viewed. A web page contains vast information in the form of text, image, audio, video, animation or hyperlinks.

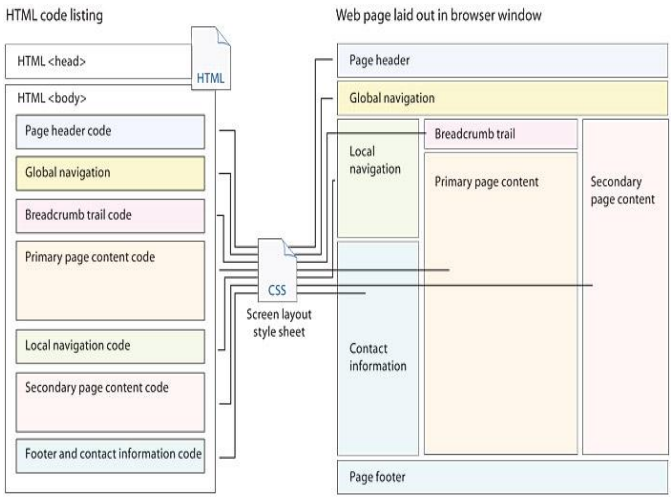

Fig.1 Source code execution model [6]

To design any web page, most important thing is to maintain document order which is the sequence of site identity, hyperlinks, primary content, related content that appears in the document source code. Aspect ratio is also an important objective to be considered in document order as the same page can be viewed in different sized devices by the end user. A web-based application is made up of web pages which are a client-server application in which user interacts

with the web pages on the web browsers. When client searches for something on web then the web browser asks web crawlers to search for the given content. Still some security approaches have been proposed like privacy can be maintained using decentralized cipher policy approach with attribute based encryption to deliver data or source code [5]. Now web crawler searches and gets result in thousands of websites but gives only few of them according to indexing. This indexing is done according to the relevant content found at the top of web page. So a web page must contain the most important information at the top in order to have a significant effect on the machine readability of web page. Like web-based applications, web browsers also have their source code [6].

B) Web browser execution engine architecture

Web browser execution environment is an environment which provides services to the clients by executing the application. Application server builds the execution environment of the web browser compliant with J2EE which is a standard specification.

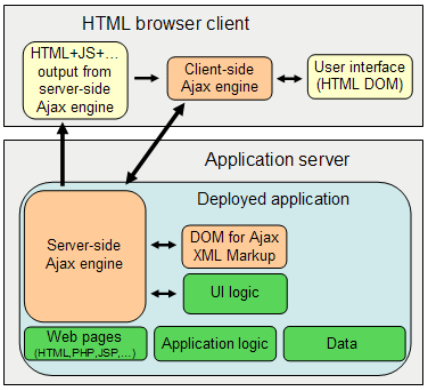

Fig. 2 Client Server model for source code execution [7]

A web server consists of three major components - WebPages, application logic and data. Besides this, server may or may not have an Ajax engine. On the other hand client-side components consist of user interface, data management and an Ajax engine with which the browser always renders revised content on the web page. Sometimes server also has an Ajax engine and performs most of the Ajax transformation. Then, client side Ajax engine directly communicates with the server side Ajax engine hiding the javascript that runs in the browser inside the widgets. The main benefit of this model is that it allows use of server side languages like java or ruby on rails for debugging, editing or refactoring tools with which developers are already known. Ajax provides several architecture approaches for various software products. Some of the Ajax runtime toolkit uses single Document Object Model (DOM) approach while some of them use dual-DOM approach [7]. Mediator concept can be applied while delivering and securing data with cryptographic approaches like attribute based encryption with decentralized or one time password approach [8].

The Single DOM approach is particularly well suited for situations where the developer is adding islands of Ajax capability within an otherwise non-Ajax DHTML application, as the programming model matches the traditional approach used in DHTML applications. Thus OpenAjax provides more efficient and greater speed to deliver data to client which makes it more beneficial [7].

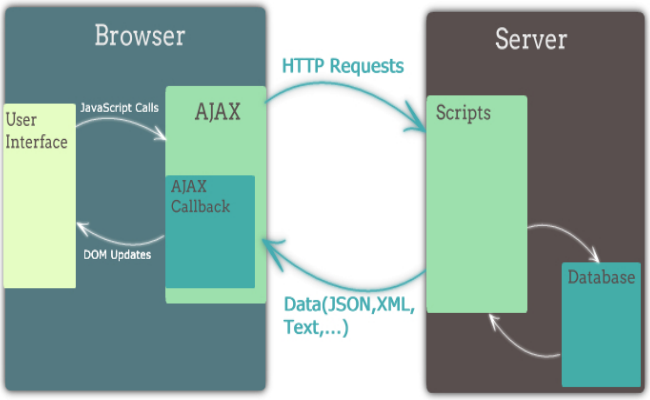

Fig.3 AJAX engine for code execution [7]

Modification in the code to improve its quality is code optimization. The main reason to produce an effective and optimized source code is that it should consume less memory space, reduces size of code, and also perform faster operation that is gives higher performance. The main thing that we should remember income optimization is that the output of code should remain same as that of un optimized source code. Optimization of code can be performed by programmer or by automatic optimizers that are either software or in-built unit in the compiler. Classification of optimization can be done into ways high level optimization and low level optimization. High level optimization is done before the conversion of high level code into machine code. It is done by the programmer itself. It includes optimization in algorithm, loop statements, removing of white spaces from the code etc. Low level optimization is performed after the conversion of code into machine language. This is achieved by the automated optimizers and performed by the in-built compilers. According to the Pareto Principle most of the execution time is spent in executing 10% of the source code [9]. So by optimizing that, 10% of the source code will reduce our developing and optimization time rather than optimizing the whole code. Software Engineer's Energy-Optimization Decision Support framework also used to help the software developer to produce more optimized code.

SEEDS provide automated analysis, decision making and implementation of those decisions in the code. SEEDS API automatically select the most energy efficient way. Code optimization increases consistency, makes code more readable, more efficient refactoring and improves the workflow of the code. It also helps in easy up gradation of code in future effectively [10].

III. PROPOSED SECURITY LOGIC

Source code is delivered in the form of binary formats. Proposed approach is directly applied on binary formatted data which is travelled along internet. It matters whether its 8/16/32/62 bit system. If it is 8 bit system then message delivered is in the 8-bit format which depends on the users system. Users bit capacity varies according to configuration of that system including processor capacity, operating system installed and so on. Proposed approach is applied to the binary formatted data. Binary formatted data is considered as base data on which cipher policy is applied.

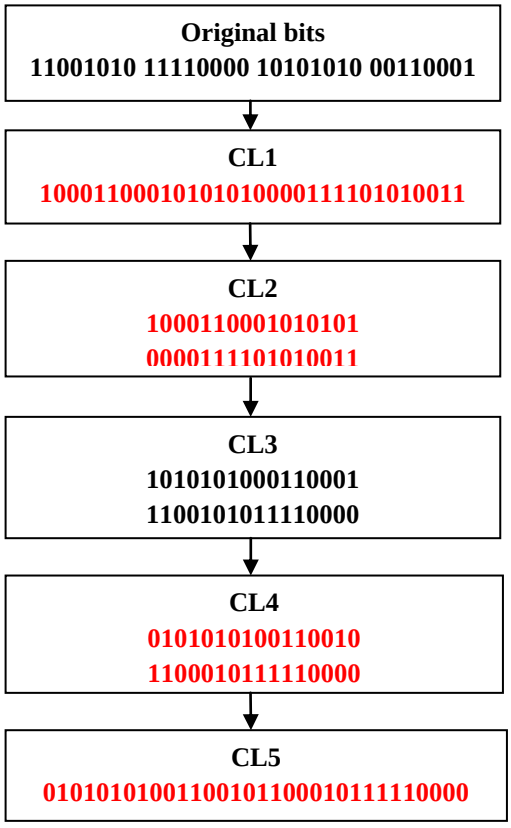

Fig.4. Proposed 5 Level cipher logic on source code

If delivered data is '11001010 11110000 10101010 00110001' which represents some login credentials, then cipher policy will be applied on it. In proposed research manuscript, logic is applied to the delivered data which can be decrypted using same logic only. Logical steps are,

- Reverse the all bits data.
- Divide the newly formed data bits into two same segments.
- Reverse the order of bits which are divided into two segments.
- Interchange two bits sequentially.
- Merge them finally to form cipher data.

In considered example, data is of 32bits system is as shown in Fig.4. Proposed five level cipher logic (CL) has been applied. First level is cipher logic 1 where original data bits are reversed. Cipher Logic 2 is applied to divide the newly formed data bits into two similar sized segments. Cipher Logic 3 is applied to reverse the order of data bits of divided segments separately. Cipher logic 4 is applied where all sequential binary bits from newly formed encrypted data will be interchanged sequentially. At the final cipher level logic 5, two segments will be merged to form encrypted data. It will be final encrypted cipher data. This encrypted data will be delivered through internet. To decrypt it, same logic is applied in reverse order to form the original data. Here even any intruder receives the encrypted data; he or she cannot understand the meaning behind the original data. Original source code is delivered through internet which is further loaded into web browser to get the required output. Instead of delivering original data, proposed logic is merged with document object model of the web browser which only understands the decryption cipher logic. It will improve the security without affecting the original content. There is no need to change other technologies or other supports for security purpose. It is purely binary security operation which is supported universally. Encryption technique can be applied independently.

IV. CONCLUSION

Web browser loads the code which generates required output. Most of the existing security techniques are applied with additional supports like platform, technology and ASCII values. Proposed approach is applied to the binary values directly which are universal. Existing security techniques are applied to the binary values but without respect to ASCII character. Even existing security techniques are applied directly to the binary data to form encrypted data where more complexity is emerged due to which efficiency and performance of the web browser is reduced. Hence in proposed approach, five leveled cipher logic is applied to enhance the security level which executes in efficient way. All approach will be applied to the whole source code of the web page which is delivered through internet.

REFERENCES

[1] J. Bethencourt, A. Sahai, B. Waters, "Ciphertext-Policy Attribute-Based Encryption", *2007 IEEE Symposium on Security and Privacy (SP '07)*, Berkeley, pp. 321-334, 2007.

[2] SU. Rasal, ST. Shelar, VS. Rasal, "*Securing Internet Banking Using Multiple Attributes Scheme And OTP",* The IIOAB Journal, Vol.7, Issue.10, pp.26-30, 2016.

[3] VS Rasal, SU Rasal, ST Shelar, "*Enhancing Privacy And Security Through Mediator Using DCP-ABE With OTP",* The IIOAB Journal, Vol.7, Issue.1, pp.277-283, 2016.

[4] S. Rasal, S. Relan, K. Saxena, "*OTP Processing using UABE & DABE with Session management*", International Journal of Advanced Research in Computer Science and Software Engineering, Vol.6, Issue.5, pp.57-59, 2016.

[5] VT. Mulik, K. Saritha, SU. Rasal, "*Privacy Preserving Through Mediator in Decentralized Ciphertext policy Attribute Based Encryption*", IJRET: International Journal of Research in Engineering and Technology, Vol.5, Issue.6, pp. 535-540, 2016.

[6] PJ. Lynch, S. Horton, "*Web style guide",* New Haven: Yale University Press, US, pp.79-177, 2016.

[7] Matthias Hertel, Yehuda Katz, Jon Gunderson, Lori Hylan-Cho, Prasanna Bale, MN. Hoyt, "*Successful Deployment of Ajax and OpenAjax*", Open AJax alliance, Vol.1, Issue.1, pp.1-6, 2015.

[8] S. Rasal, M. Matta, K. Saxena, "*OTP system with third party trusted authority as a mediator*", International Journal Of Engineering And Computer Science, Vol.5, Issue.5, pp.16566-16568, 2016.

[9] R. Sanders, "*The Pareto principle: its use and abuse*", Journal of Services Marketing, Vol.1, Issue.2, pp.37-40, 1987.

[10] I. Manotas, L. Pollock, J. Clause, "*SEEDS: a software engineer's energy-optimization decision support framework",* In Proceedings of the 36th International Conference on Software Engineering, New York, pp.503-514, 2014.

Authors Profile

Mr. Suraj U. Rasal pursed Master of Science in Computer Science from University of Bedfordshire, England and Bachelor of Engineering degree in Information Technology from University of Mumbai. Now he is working as Assistant Professor at Bharati Vidyapeeth University College of Engineering Pune. He has published fourteen research papers since doiing research work. He is also working as a reviewer for some Scopus indexing journals. For more info Google scholar link: https://scholar.google.co.in/citations?user=nTZcfh8AAAAJ&hl=en&oi=ao

Mr. Robin Redhu is pursuing Under Graduation in Computer Engineering stream from Bharati University College of Engineering Pune. He is doing his research work in Network Security and cryptography since last two years.

Mrs.Varsha S Rasal received the Bachelor of Engineering degree from Anna Univeristy,Tamilnadu,India in Computer Science and Engineering and Masters in Computer science and Engineering from Calicut University, Kerala, India. She has published nine research papers in international journals and conferences in the same domain since doing research work.

Mrs. Shraddha T Shelar received the Bachelor of Engineering degree in Information Technology branch and Master in Technology degree from University of Pune,India. Now she is working as Assistant Professor in D Y Patil College of Engineering Akurdi,Pune. She has published seven research papers in international journals and conferences in the same domain since doing research work. All details are available on google scholar account.

