Transactions on Engineering and Computer Science

Research| Vol 1 Iss 1

String Manipulation in Random Order by Array Index in Hypertext Preprocessor

Danial Kafi Ahmad[1,2*], Zul Hilmi Abdullah[1], and Mariam Farida Ahmad[3]

[1]Faculty of Information Technology, INTI International University, Nilai, Malaysia

[2]Language Centre, National Defence University of Malaysia, Kuala Lumpur, Malaysia

[3]Information Management Division, Department of Statistics, Malaysia

***Corresponding author:** Ahmad DK, Faculty of Information Technology, INTI International University, Nilai, Malaysia; Language Centre, National Defence University of Malaysia, Kuala Lumpur, Malaysia, Tel: 067982000; E-mail: danialkafi.ahmad@newinti.edu.my

Received: May 30, 2020; **Accepted:** June 30, 2020; **Published:** July 10, 2020

Abstract

Encryption is the process of converting the initial text into different text expression, whereby it could protect the original text from exposed to an unauthorized entity. Most of the software-based system would implement the mechanism so that it could protect the data. The need for understanding the variety encryption and their related concepts seems vital as it helps developer to produce better technique and design of encryption, and one of it is string manipulation. In addition, more variety of design could be tested rigorously so that it would might produce relevant important findings for group of people who involved in software security either directly or indirectly, by knowing the possible limitations and advantages in any particular designs. In this research experiment, a light string manipulation module are developed using scripting language of both Hypertext Preprocessor (PHP) and Hypertext Markup Language (HTML).

Keywords: ***Text, Array, Encryption, Random.***

1. Introduction

Every software-based system would require the security mechanism to be deployed in within the system. For example, the implementation of encryption which could convert the plaintext to unreadable and random text by means of automation [1]. However, those technique would have been derived from so called “simple and manual technique” which own limited capability in terms of their efficiency (capacity) and computation’s result, which then transform into automated mechanism for computation. This shows that a basic variety of algorithm or solution were always been

Citation: Ahmad DK, Abdullah ZH, and Ahmad MF. String manipulation in random order by array Index in hypertext preprocessor. Trans Eng Comput Sci. 2020;1(1):108.

important and would need to be developed consistently regardless of the success current technique of encryption, so that it could be enhanced or improved for future used. The good mechanism of encryption would be very important to be deployed in most of the online business platform, of Web Based system (E-Commerce) which are widely used across all regions, and this also reflects the importance of scripting language which were used to develop the system. One of the languages which are accessible easily is the Hypertext Preprocessor (PHP) and Hypertext Markup Language (HTML) as they are of the open source type. The understanding of the encryption mechanism was supposed to be aligned with the understanding of specific scripting language (PHP & HTML), as the mechanism could be interpreted in different manner by other language in terms of their flow of execution. In addition, it is vital to understand deeply and exploring the encryption concept in PHP and HTML. Therefore, in this research experiment, a light module of encryption that able to manipulate the original text (user input) by using array concept and random number were developed in order to investigate the ability of built in functions in PHP and HTML to perform basic encryption. The developed light module was to focus on the concept and mechanism that could be used to for any possible extension or learning rather than the usage in commercial context.

2. Prior Work

As an increasing number of applications transitions to the Web, the need of ordinary users to have more secure Web applications had increased and Web developers are attempting to meet those expectations. They need to fulfill the users' expectation in providing secure web application due to increasing numbers of critical applications such as online banking, e-commerce system, e-learning, and other related applications [2]. Those systems required data protection in terms of confidentiality, integrity, and availability. The cryptography techniques can be applied for data confidentiality and integrity protection. Commonly, proper backup and recovery techniques will be used for availability protection. There are various techniques proposed to enhance web application security such as code obfuscation techniques which randomize the web application source code to make the code harder to read [3]. However, more complex code will decrease the performance of the application itself. There were more works proposed to utilized the cryptography techniques includes the established encryption with algorithms in securing web applications [4]. The researcher applied the cryptography techniques in various layers of protections such as source code layer [5,6], password, or authentication layer [7,8], database layer [9,10] as well as a communication layer [11]. Although various solutions were already proposed, there are still improvements and enhancement required due to the increasing number of new types of attacks towards web applications. One of the main concerns in data protection is to balance the effectiveness of protection measures and application performance. This research tries to propose lightweight solutions that can be embed with existing solutions that will contribute minimal impact to the application performance.

3. Objectives

The objectives of this research were to develop a light string manipulation module using the scripting language which are PHP and HTML. The development of this module would be able to increase the understanding towards the concept of string manipulation which much related to cryptography studies.

4. Methods

4.1 SDLC

The non-plan driven technique was implemented in this program development as opposed to the systematic approach promoted in within Software Engineering field. The program was considered as small-scale project as it comprises of few lines of codes of PHP and HTML. The Non-Plan driven promotes flexibility during development, in terms of response to adhoc requirement changes. However, it consists of few models which called as software process model, whereby incremental model was deployed in this development. Incremental model comprises of few activities such as Specification (Requirement), Development (Coding) and Validation (Testing).

4.2 Software process model (Incremental)

4.2.1 Specification

4.2.1.1 Development tools

Apache were required in this research experiment in order to execute the PHP scripts, whilst the client (Browser) were required to display the GUI of HTML scripts. The Apache were installed via XAMPP package (v3.2.2) of *Cross Platform* which available as an open source tool. Notepad application were used as a platform for the program scripts to be written and were stored in htdocs folder of XAMPP in within C: partition.

4.2.1.2 Architecture-Client server

The developed program are in the form of Web Based whereby the Client Server architecture were deployed in the system. The architecture allows the exchange of information (request and response) in between Client (Browser) and Server (Apache of XAMPP) [12]. A page or program were requested via the Uniform Resource Locator (URL) with the configured port (8080).

4.2.1.3 Pseudocode

In this development, pseudocode were developed in order to design the system flow as well as to visualize the program mechanism [13]. Fig. 1. shows the pseudocode of the service page (enc3.php).

```
BEGIN

   IF the input is not empty
      then $w[rand(0,strlen($w=input from user-1)]=generate random no;
           Display manipulated string($w)
                                    ELSE
    then Display "No data inserted..."Redirect to main page (inputenc.html)

END
```

Fig. 1. Pseudocode of service page (enc3.php)

4.2.2 Development

4.2.2.1 Code

The program were developed using the components as shown in Table 1. It also contains the description and purpose of each function and statement.

Table 1. Purpose of program components.

Function/Statement	**Purpose**
<?php ?>	Indicating the start and end point of the PHP program
echo()	Display output
strlen()	Return the number of counts of string
$w	A variable which hold the string data type keyed in by the user as well as overwritten string after manipulation
$_POST[]	Retrieve the data from the textbox input of HTML type
rand()	Return a random number (integer) in within a certain range
empty()	Validate input
if()else{}	Validate condition of input
-1	Required in order to get the exact index number of the string (keyed in data)
$w[]	Defining and overwriting character at

	specific index (offset)
header()	Redirect to required page
<html></html>	Indicating the start and end point of HTML program
<form></form>	Indicating the start and end point of form of HTML
method="post"	Retrieve data
action="enc3.php"	Redirect to enc3.php webpage
name=""	Define name for the input
value=""	Define text on the input type (*submit type)
<input type="text">	A graphical textbox to accept input
<input type="submit">	A graphical button to transmit data

In this research experiment, the flow chart were used as a tool to design and depict the system flow in addition to pseudocode [14]. In some instance, flow chart were assume to be more friendly as compared to pseudocode whereby it uses graphical images (shapes) to indicate the program's flow. Whilst, flowchart were much closer to the executed program's script as compared to flow chart. However, both were interrelated whereby a flowchart could be transformed from a pseudocode. Fig. 2. shows the components of flow chart.

Shape	Action
	Process/Computation

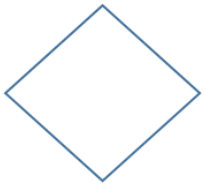
	Input/Output
	Condition/Decision
	Flow/Sequence/Direction

FIG. 2. Components of Flow Chart.

4.2.2.2 Installation and configuration

A configuration were required in this development as the default port were unavailable. The port were change to 8080 and this caused the request made via URL were embedded with the new port number. Configuration (Listen 8080) were made via the *httpd.conf* file which retrievable from the XAMPP Control Panel. In some instances, the services such as VMWARE should be terminated in order to allow the port availability. Figs. 3 and 4 shows the Graphical User Interface (GUI) of httpd.conf and services list in XAMPP

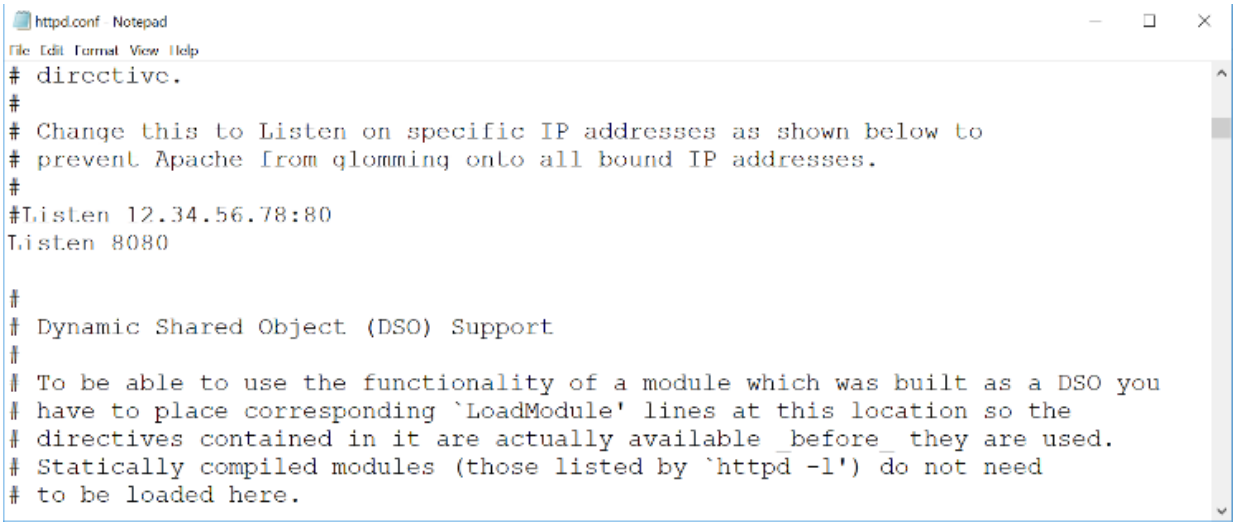


Fig. 3. Configuration steps of port via *httpd.conf* file.

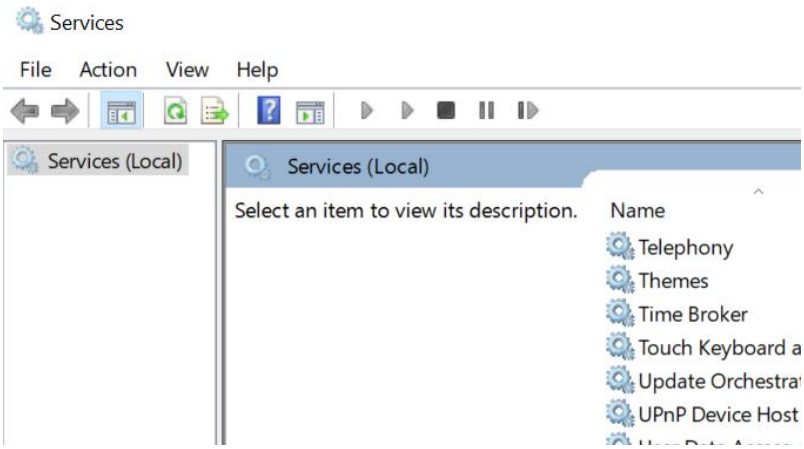


Fig. 4. GUI of services list via XAMPP.

4.2.3 Validation

Required activities in testing were conducted towards the program scripts in order to achieve the Validation and Verification (V&V) [15]. The following test level and type were conducted towards the program script

4.2.3.1 Test Level

Fig. 5 shows the test level model which were carried out in the research experiment.

Unit (Functional, Non-Functional)->Static, Dynamic

Integration (Functional, Non-Functional)-> Static, Dynamic

System (Functional, Non-Functional)->Static, Dynamic

Acceptance (Functional, Non-Functional)-> Static, Dynamic

Fig. 5. The test levels.

Table 2 shows the template of the test case that were used to test the Functional and Non-Functional type of testing. Test Case were very important in order to represent the purpose of the testing that were conducted in the research experiment.

Table 2. Test Case Template.

Test Case ID	***The unique ID***
Type	*The test type (Functional or Non-Functional)*
Objective	*The purpose of testing*
Input	*The input (String type etc.)*
Procedure/Steps	*List of actions required*
Expected Output	*The expected displayed results prior to execution*
Actual Output	*The displayed results of execution*
Result	*Test case results*

5. Result and Discussion

5.1 Algorithm

5.1.1 Coding

```
<html>
<form action="enc3.php" method="post">
<input type="text" name="input">
<input type="submit" value="compute">
</form>
</html>
```

Fig. 6. Main page (inputenc.html).

Fig. 6. shows the script of HTML type which were written in within the notepad file of HTML type and extension. <form> tag of HTML were used in this program to display the Graphical User Interface (GUI) of textbox field and submit button. In within the <form> tag, action=”enc3.php” were used in order to redirect the user to other web page (enc3.php) with keyed in data (user input. The data transmission were done using the method POST which were written as method=“post”, in within the <from> tag. The <input> of type text were used in this program as to accept input of string type from user dynamically.

```
<?php
If (!empty($_POST['input'])){$w[rand(0,strlen($w=$_POST['input'])-1)]=rand(0,9);
echo $w;}
else {echo "No data inserted...";}
header("refresh:1; url=inputenc.html");
?>
```

Fig. 7. Service page (enc3.php).

Fig. 7. shows the service page (enc3.php) whereby in this program, the computation of replacing the single character of string (user input) keyed in occur. For example, if the user would keyed in the string “abc”, it would be then counted as 3 characters and this include space. The character counting were computed using the function strlen() in strlen($w=$_POST['input'])-1. The argument in the strlen() function which is $w=$_POST['input'] would retrieve the data from the HTML form and stored in variable $W. For example, the count abc (input) consist of 3 characters, therefore the strlen () would return the value of 3, however, the $w would hold a value of 2 due to the subtraction of -1 which is 3-1=2. The -1 were required in order to get the range of index i.e. (0,1,2) as to create order of alphabets replacement according to index. For example, consider Fig. 8. scripts and its elaboration.

*Assuming following condition in the first run:

- rand(0,9) produce 8.
- $_POST['input'])hold *abc*

- rand(0,strlen($w=$_POST['input'])-1) produce 0.

```
$w[rand(0,strlen($w=$_POST['input'])-1)]=rand(0,9);
$w[rand(0,3-1)]=8;
$w[rand(0,2)]=8;
$w[2]=8;
```

Fig. 8. Elaboration of array redefinition of specific offset.

The statement after the elaboration would be $w[2]=8 which means the variable $w which initially hold value of string type (abc) would be then replace by the value of 8 of alphabets with index or offset 0. This means that the string (abc) which has index of (*0→a, 1→b, 2→c*) would then become 8bc as the final string. echo () statement then were used to display the manipulated string as an output on the screen. In prior to the string manipulation computation, a control structure of condition has been deployed in order to validate the whether data has been keyed or not. If not, no computation occur and the page would redirect to main page from service page. Fig. 9 shows the design of the proposed manipulation technique in the form of flow chart model.

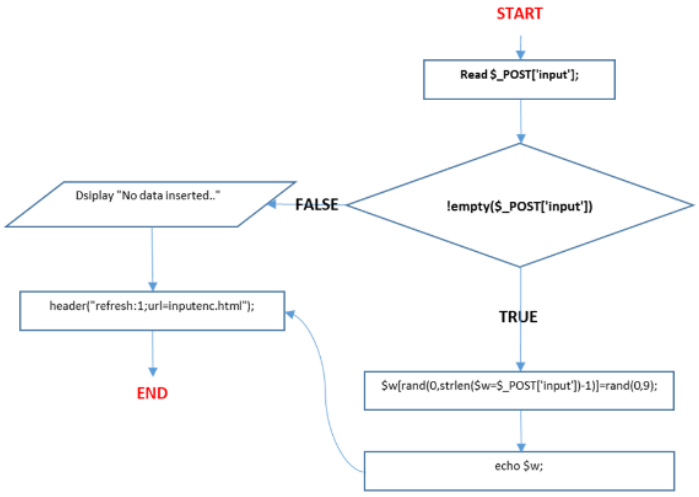


Fig. 9. The flow chart model of the encryption module.

5.1.2 Graphical user interface

Figs. 10, 11, 12, and 13 shows the output of all the versions of the program.

Fig. 10. Output of main page.

Fig. 11. Output of main page with (abc) as an input.

ab8

Fig. 12. Output sample of a first run with input (abc).

a9c

Fig. 13. Output sample of another run with the same input (abc).

5.2 Testing (Validation)

Functional and Non-Functional Requirement type of the program were tested and carried out as an activities in testing as to make sure the working functionality of the program. However, the main concern in this research experiment were the Functional Requirement whereby the manipulation computation were tested in two runs. The testing were based on own defined exit criteria i.e. (Web page linking, user input, data manipulation computation, and response time). Below shows the test cases of few testing types which were conducted on each of the test level, which then led to the acceptance of the system.

5.2.1 Unit Testing

Table 3, 4, 5, 6, 7, and 8 shows the test cases of tested Functional and Non-Functional requirement at unit test level.

Table 3: F1 Test Case.

Test Case ID	**F-1**
Type	Functional
Objective	To test the functionality of text field of *inputenc.html* program
Input	String Type Data
Procedure/Steps	I. Enter data II. Click submit button
Expected Output	Input with single alphabet replaced by new value (ab8)
Actual Output	ab8
Result	PASS

Table 4: F2 Test Case.

Test Case ID	**F-2**
Type	Functional
Objective	To test the functionality of data manipulation of *enc3.php* program with similar output
Input	String Type Data (similar as to previous run)
Procedure/Steps	I. Enter data

	II. Click submit button
Expected Output	Input with single alphabet replaced by new value (a9c)
Actual Output	a9c
Result	PASS

Table 5: F3 Test Case.

Test Case ID	**F-3**
Type	Functional
Objective	To test the functionality of display statement of *enc3.php* program
Input	String Type Data
Procedure/Steps	I. Enter data II. Click submit button
Expected Output	Input with single alphabet replaced by new value (a9c)
Actual Output	a9c
Result	PASS

Table 6: NF1 Test Case.

Test Case ID	**NF-1**
Type	Non-Functional
Objective	To test the response time of text field displayed of *inputenc.html* program
Input	String Type Data
Procedure/Steps	I. Enter data II. Click submit button
Expected Output	Keyed in data appeared instantly in the textbox
Actual Output	Keyed in data appeared instantly in the textbox
Result	PASS

Table 7: NF2 Test Case.

Test Case ID	**NF-2**
Type	Non-Functional
Objective	To test the processing time of data manipulation of *enc3.php* program with similar output
Input	String Type Data (similar as to previous run)
Procedure/Steps	I. Enter data II. Click submit button

Expected Output	Input with single alphabet replaced by new value (a9c) displayed in within the timeline as in NF-1
Actual Output	a9c
Result	PASS

Table 8. NF3 Test Case.

Test Case ID	**NF-3**
Type	Non-Functional
Objective	To test the processing time of display statement of *enc3.php* program
Input	String Type Data
Procedure/Steps	I. Enter data II. Click submit button
Expected Output	Input with single alphabet replaced by new value (a9c)
Actual Output	a9c
Result	PASS

5.2.2 Integration and System Testing

Table 9, 10, 11, and 12 shows the test cases of tested Functional and Non-Functional requirement at integration and system testing test level.

Table 9: F1-ISY Test Case.

Test Case ID	**F-1-ISY**
Type	Functional
Objective	To test the linking in between *inputenc.html* and *enc3.php*
Input	String Type Data
Procedure/Steps	I. Enter data II. Click submit button
Expected Output	enc3.php page displayed along with data keyed in (abc)
Actual Output	ab8
Result	PASS

Table 10: F2-ISY Test Case.

Test Case ID	**F-2-ISY**
Type	Functional
Objective	To test the linking in between *inputenc.html* and *enc3.php* via header()
Input	String Type Data
Procedure/Steps	I. Enter data

	II. Click submit button III. Wait for 1 second after *enc3.php* page
Expected Output	inputenc.html displayed
Actual Output	nnputenc.html displayed
Result	PASS

Table 11: F3-ISY Test Case.

Test Case ID	**F-3-ISY**
Type	Non-Functional
Objective	To test the linking processing time in between *inputenc.html* and *enc3.php*
Input	String Type Data
Procedure/Steps	I. Enter data II. Click submit button
Expected Output	enc3.php page displayed along with data keyed in (abc) in less than a second
Actual Output	ab8
Result	PASS

Table 12: F4-ISY Test Case.

Test Case ID	**F-4-ISY**
Type	Functional
Objective	To test the linking processing time in between *inputenc.html* and *enc3.php* via header()
Input	String Type Data
Procedure/Steps	I. Enter data II. Click submit button III. Wait for 1 second after *enc3.php* page
Expected Output	inputenc.html displayed in less than a second
Actual Output	nnputenc.html displayed in less than a second
Result	PASS

6. Conclusions

Data manipulation were vital in any system available today. Without data manipulation, it would be assumed that a system were of lesser security aspects. This is due to the nature of the system especially the web based whereby the exchange of information occur via data transmission in within the Commercial Interconnected Network (Internet). In some of the security threats and attacks, those were from the data hijacked during transmission [16]. Therefore, it is

necessary for the data to be protected prior to transmission and data manipulation is one of the key to achieve this goal. In other words, it could be as so called “encryption” whereby the data (text input) is being changed to other data (text) of different meaning or expression. As for many system available today, the design of the security may varied and the reliability could be varied based on the design. It would be something like “the better the design, the better the security, the better is the reliability”. And in many security mechanism, it always a good idea to go into details, and in this research context, is the data manipulation, and these could be achieved via research etc. Therefore, it is important to develop a variety of design with variety of mechanisms in terms of data manipulation computation. It is strongly recommended that the researchers in computing and any field to involve more consistently and regularly in the research which related to data manipulation as an aspect of security in computing system.

7. Future Enhancement

The developed module were seems to be useful in understanding the basic concept of data manipulation as a security aspect in a context of scripting languages specifically in PHP and HTML, and also to provide awareness of the data manipulation with their importance. It has few aspects that could be improved and extended in order to be used in commercial context. For example, the complexity of single character manipulations could be enhanced in the future by having multiple character manipulation. In addition, the value generated could be enhanced by the multiple alphabets generated with special character, and the module is not meant and solely focus on data during transmission but rather the manipulation of string technique.

8. Acknowledgment

The authors would like to express their gratitude to the Office of Research Development & Consultancy of INTI International University (IU) and the Faculty of Information Technology INTI IU for the supports in terms of resources allocation for the research. The main author would like to show his gratitude to Assoc. Prof. Dr. Norshima Zainal Shah and Dr. Hoo Yann Seong of National Defence University of Malaysia for the priceless supports and motivation in doing research. Finally, the authors would like to express their gratitude to Dr. A. Selamat, former associate researcher at the Institute for Mathematical Research (UPM) for important advice on computational logic.

REFERENCES

1. Gupta H, and Sharma VK. Multiphase encryption: A new concept in modern cryptography. Int J Compu Theory Eng. 2013;5(4):638.
2. Cairns K, Halpin H, and Steel G. December. Security analysis of the W3C web cryptography API. In International Conference on Research in Security Standardisation, 2016, 112-140p.
3. Maskur M, Sari Z, and Miftakh A. Implementation of obfuscation technique on PHP source code. Proc Elect Eng Comp Sci Informat. 2018;5(1):738-742.
4. Widjaja A, and Mujito CS. Data security applications with a combination of cryptographic methods and tiny encryption algorithm twofish web based on pt jakarta sidola computer. Int J Pure AP Mat. 2017;117(15):905-914.

5. Watt C, Renner J, Popescu N, et al. Ct-wasm: Type-driven secure cryptography for the web ecosystem. Proc the ACM on Programming Languages, 2019 3(POPL), pp.1-29.
6. Marashdih AW and Zaaba ZF. Cross site scripting: removing approaches in web application. Procedia Comput Sci. 2017;124:647-655.
7. Sumagita M, Riadi I, Sh JPDS, et al. Analysis of secure hash algorithm (SHA) 512 for encryption process on web-based application. Int J Cyber-Sec Digit Forensics. 2018;7(4):373-381.
8. Ajayi OO, and Fanala TF. Empirical evaluation of data hashing algorithms for password checks in PHP webapps using salt and pepper. Comput Inf Syst Dev Inf Allied Res J. 2017;8:21-28.
9. Sethi S, and Singhal V. ICTS2016-SS27-07: A Peek into Web Applications Security. In Proc Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies, 2016, 1-3p.
10. Lorey K, Buchmann E and Böhm K. TEAL: Transparent encryption for the database abstraction Layer. In CAiSE Forum, 2016, 145-152p.
11. Lescisin M. and Mahmoud Q. Tools for active and passive network side-channel detection for web applications. In 12th {USENIX} Workshop on Offensive Technologies ({WOOT} 18), 2018.
12. Civanlar MR, and Haskell BG. Client-server architecture using internet and public switched networks. U.S. Patent 5,995,606, 1999.
13. Roy GG. Designing and explaining programs with a literate pseudocode. J Educ Res Comput. 2006;6(1),1-es.
14. Xinogalos S. Using flowchart-based programming environments for simplifying programming and software engineering processes. In IEEE Global Engineering Education Conference, 2013, 1313-1322p.
15. Spillner A, Linz T, and Schaefer H. Software testing foundations: a study guide for the certified tester exam. Rocky Nook, Inc, 2014.
16. Feng J, Lu G, Wang H, et al. Supporting secure spectrum sensing data transmission against SSDH attack in cognitive radio ad hoc networks. J Netw Comput Appl. 2016;72:140-149.

Authors Biography

Danial Kafi Ahmad received the MSc. degree in advanced software engineering from the University of Leicester, England, United Kingdom and the B.C.Sc. in software engineering from the National Energy University, Malaysia. He is currently a Lecturer with the INTI International University & Colleges. He is also a holder of Tester's Certificates (CTFL-AT and CTFL, from the Australian & New Zealand Testing Board and Malaysian Software Testing Board respectively). He has a strong interest towards studying the theory of programming codes and their related field.

Zul Hilmi Abdullah received the MSc degree in computer science-information security from the Universiti Teknologi Malaysia and bachelor of computer science degree and diploma in computer science from Universiti Putra Malaysia.. He is currently serving as a Lecturer cum Head of Department with the Faculty of Information Technology, INTI International University.

Mariam Farida Ahmad received Bachelor in Computer Science (Hons) from Universiti Teknologi MARA. She is currently working as a Government Servant with the Department of Statistics in Malaysia.

Citation: Ahmad DK, Abdullah ZH, and Ahmad MF. String manipulation in random order by array Index in hypertext preprocessor. Trans Eng Comput Sci. 2020;1(1):108.