Project Documentation

AND

Test Cases

For

<u>Triple-DES Algorithm Using Java Script</u>

PART - 1

Documentation

1: Introduction

The Triple-DES Simulation is used to clarify the working of Triple Des algorithm using the DES algorithm.

Languages Used:

HTML and CSS

JavaScript

2:Overall Description

2.1)- Part_1:

This part is used to generate the plain text in binary format and two keys in hex format. The field are generated using random function of Java-Script.

2.2)- Part_2:

This is the part where you provide inputs according to the manual and proceed with the encryption and decryption.

2.3)- Part_3:

This is the part where the status is shown which verifies whether the steps followed are correct or not and output the corresponding mistake.

3 : Working Description

Here functionalities of all the functions defined inside the program is discussed.

Functions are listed below:

form 64 bit binary key

key PC 1

key PC 2

shift left 1

key per round

key init

remove spaces

IP text

text init

XOR F

text Round Encrypt

FP text

E Prem text

P Prem DES function

Calculate S Box

DES F function

var init

Encrypt

Decrypt

text Round Decrypt

check Answer

Triple Des Decrypt

Triple Des Encyrpt

Des Decryp

Des Encyrpt

change Plain text

change key A

change Key B

3.1)-form 64 bit binary key:-

This function is used to convert the given key in hexadecimal value to binary value.

3.2)-key PC 1:-

This function is used to permute the given text and generate a 56 bit key from 64 bit one.

3.3)-key PC 2:-

This function is used to permute the key with PC2 and generate the key for round.

3.4)-shift left_1:-

This function bitwise and circular shift the bits of key passed by parameter by one.

3.5)-key per round-:

This function is used to generate the round keys for all the 16 rounds and store it in global variable.

3.6)-key init:-

This function is wrapper for 3.1 , 3.2 and 3.5

3.7)-remove spaces:-

This function removes any spaces in between the text provided to it.

3.8)-IP text:-

This function is used for doing the initial permute on the text and generate the two equal parts and store it in global variable.

3.9)- text init:-

This function is used as a wrapper for 3.7 and 3.8.

3.10)-XOR F:-

This function is used to bit wise xor the two inputs provided to it.

3.11)-text round encrypt:-

This function is used as a wrapper for DES F Function and XOR F and driving the rounds one by one.

3.12)-FP text:-

This function is used for doing the final permute on the text.

3.13)-E Prem text-:

This function is used for doing the expansion permute on the text and convert 32 bit text t 48 bit.

3.14)-P Prem DES function:-

This function is used foe permuting in the DES function.

3.15)- Calculate S Box:-

It is used for degrading 48 bit to 32 bit text . It uses the sboxes to generate and replaces the text.

3.16)- DES function:-

It is used as a wrapper for 3.13, 3.10,3.15 and 3.14.

3.17)-var init:-

It is used for reinitializing the global variables.

3.18)-Encrypt:-

This function is whole sole responsible for doing the encryption and is used as a full wrapper of 3.17, 3.9, 3.6, 3.11 and 3.12

3.19)-Decrypt:-

This function is whole sole responsible for doing the decryption and is used as a full wrapper of 3.17, 3.9, 3.6, 3.11 and 3.12.

3.20)-text round decrypt:-

This function is used as a wrapper for DES F Function and XOR F and driving the rounds one by one for decryption.

3.21)- Check Answer:-

this function is used to update the status of status field.

3.22)-Triple DES Encrypt:-

Used as a utility for 3.21 for encrypting with the correct procedure of triple des.

3.23)-Triple DES Decrypt:-

Used as a utility for 3.21 for decrypting with the correct procedure of triple des.

3.24)-DES encrypt:-

This is a on click listener for Encrypt button and uses the encrypt function for encryption.

3.25)-DES decrypt:-

This is a on click listener for decrypt button and uses the decrypt function for decryption.

3.26)-change plain text:-

This function is generating the random string of 64 bit binary number and putting it in readonly input of part 1.

3.27)-change key A:-

This function is generating the random string of 16 bit hex number and putting it in readonly input of part 1.

3.28)-change key B:-

This function is generating the random string of 16 bit hex number and putting it in readonly input of part 1.

4 : Project Quiz

Here the discussion is all about the dynamic quiz created independently created for MIPS Parser project.

It Contains following functions:-

- generate Question Container
- generate Result Container
- put containers
- put Result
- remove child
- generate Random Index
- get Content
- put Content
- check Answers
- submit Answers
- start Quiz
- submit Quiz

4.1)-generate Question Container :-

Function Introduction:-

This function is used to generate the container structure for each question and their options dynamically.

Function Implementation:-

Function takes the Question Id and Answer ID as parameter to assigned to respective containers.

4.2)-generate Result Container :-

Function Introduction:-

This function is used to generate the container structure for Result for each question and their correct and user Answers dynamically.

Function Implementation:-

Function takes the Result ID as parameter to assigned to respective containers.

4.3)- put Containers :-

Function Introduction:-

This function is used as wrapper over 4.1 and 4.2

Function Implementation:-

This function is randomly generates the number of container and then creates the that number of containers using function 4.1 and 4.2 (discussed above).

4.4)- put Result :-

Function Introduction:-

This function is used to display the results for the quiz

Function Implementation:-

Function first creates variables used as ID for different sections. After that the User Answers (stored in global variable) is used to decide the text color and result status of particular question.

4.5)- remove Child :-

Function Introduction:-

This function is used to remove the child of Result and Quiz container to make a fresh start using the DOM property.

4.6)- generate Random Index :-

Function Introduction:-

This function is used to generate the Random number within a specified range using the JavaScript inbuilt function.

Function Implementation:-

This function uses the constant 11, which is the total number of question stored as JSON data to generate the Random Number. It then check , whether the number generated is generated earlier or not and stores and regenerate accordingly.

4.7)- get Content :-

Function Introduction:-

This function is used to place the options of a passed Question in random order.

4.8)- put Questions :-

Function Introduction:-

This function is used as wrapper over 4.7 and is used to put all the questions as the number of container.

4.9)- check Answers :-

Function Introduction:-

Functions used to validate and stores the score of User for their answers.

Function Implementation:-

This function is checking whether the user has skipped the question or not and if not skipped, it stores which option is selected by user and increase the score of user , if selected option is the right answer. It returns the above status to the caller function for storing purposes.

4.10)- submit Answers :-

Function Introduction:-

This function is used as wrapper over 4.9

Function Implementation:-

This function stores the status from function 4.9 and push it on global variable for further calculations.

4.11)- start Quiz :-

Function Introduction:-

This function is on-click listener of star button of quiz

Function Implementation:-

Functions alters the visibility of some containers and their content and places random number of question with randomize option. It also removes the previous questions or result (if exists).

4.12)- submit Quiz :-

Function Introduction:-

This function is on-click listener of submit button of quiz

Function Implementation:-

Functions alters the visibility of some containers and their content and places result with their result status. It also initializes all the global variables for the fresh start..

PART - 2

Test Cases

Input_1 (Wrong Input) (20)

Execute_1 (Wrong Execution) (20)

Test Scenario ID	Wrong Input	**Test Case ID**	Input-1
Test Case Description	User provide wrong input	**Test Priority**	High
Pre-Requisite	NA	**Post-Requisite**	NA

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Browser**	**Test Result**
1	Launch application	Open newindex.htm	Home page	Home page	ff-66.0.4	Pass
2	Provide wrong input	Fill wrong input in part -2 fields	Alert with showing the warning	Alert with showing the warning	ff-66.0.4	Pass

Test Scenario ID	Wrong Order of Execution	**Test Case ID**	Execute-1
Test Case Description	User follow wrong procedure	**Test Priority**	High
Pre-Requisite	Inputs are required in part-2 fields	**Post-Requisite**	NA

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Browser**	**Test Result**
1	Launch application	Open newindex.htm	Home page	Home page	ff-66.0.4	Pass
2	Provide input	Fill input in part -2 fields	Inputs get saved	Inputs get saved	ff-66.0.4	Pass
3	Follow wrong procedure	Wrong order of decrypt and encrypt and then click check answer	Status will be procedural erro	Status will be procedural erro	ff-66.0.4	Pass

END OF DOCUMENTATION

References:-

Software Testing Help Website

Source Code of Triple DES USING JS

Prepared By:-

Yash Kudesia

Project Details:-

Project Domain – CryptoGraphy

Issue Number – 214

Project Topic – Triple DES

Submitted to:-

SRIP@IIITH