Practical Assignment 2 (PA2)

Name: Ashish Kumar
Roll No: 18075068
Dept: CSE B.Tech. 4th Year
Course: Network Security (CSE-537)

Practical Assignment 2

Perform experiments to explore the Avalanche Effect progression across the DES rounds. Use:

(i) 5 different plaintexts
(ii) 5 different Hamming distances (HD)
(iii) 5 different secret keys.

Report plots of HD against round number.

GitHub link (Source code): https://github.com/krashish8/netsec-assignments

Deployed (View Online!!): https://ashish-netsec.netlify.app/ [Alternate Link]

(Select "Practical Assignment 2" in the above link)

Note: Screenshots of the 3 subparts are attached at the end.

[P.T.O.]

Source Code:

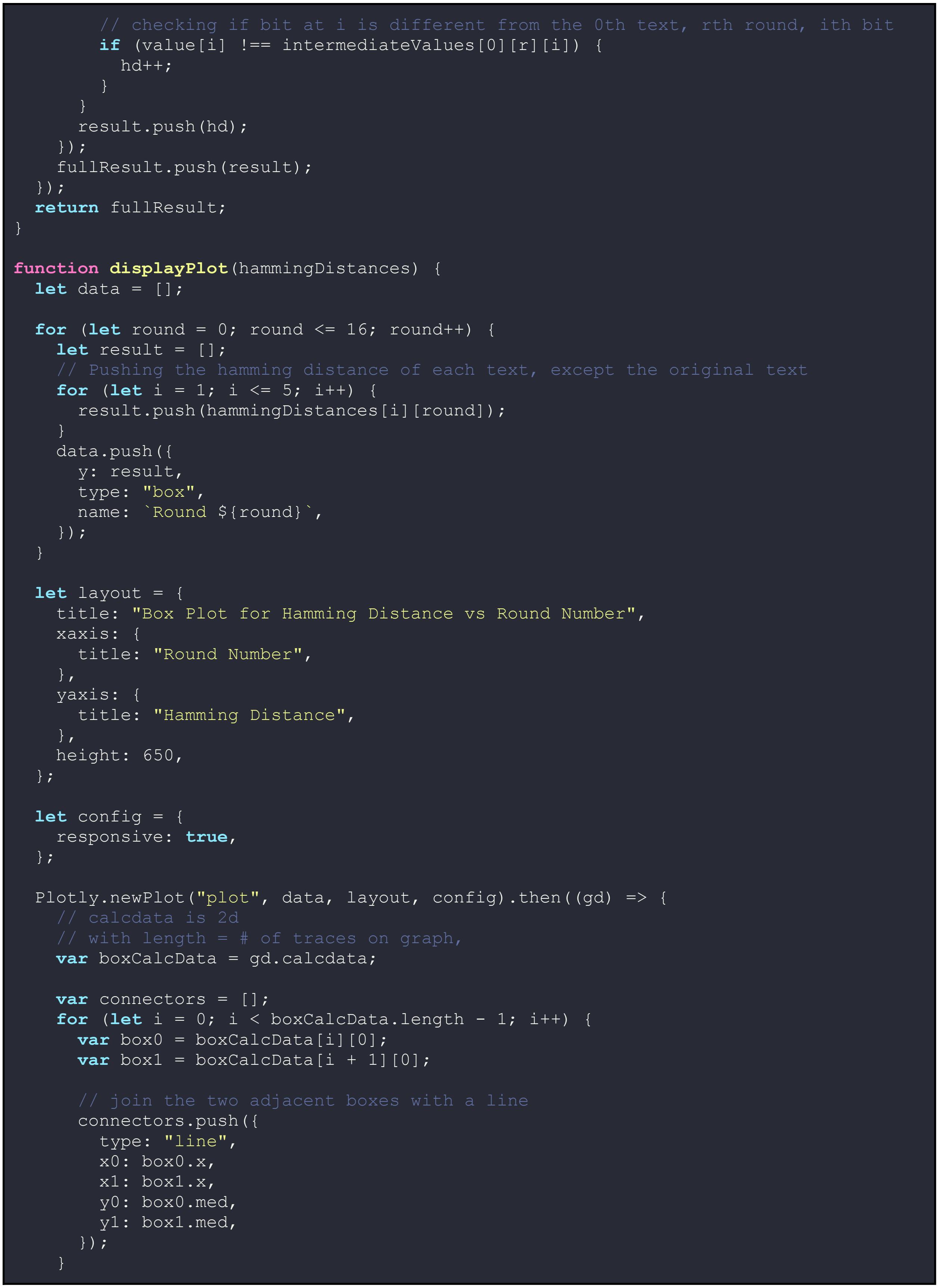
The complete source code for rendering the HTML and creating the UI can be found on the above GitHub repository.

The DES code written in JavaScript (des.js) has been adapted from this link.

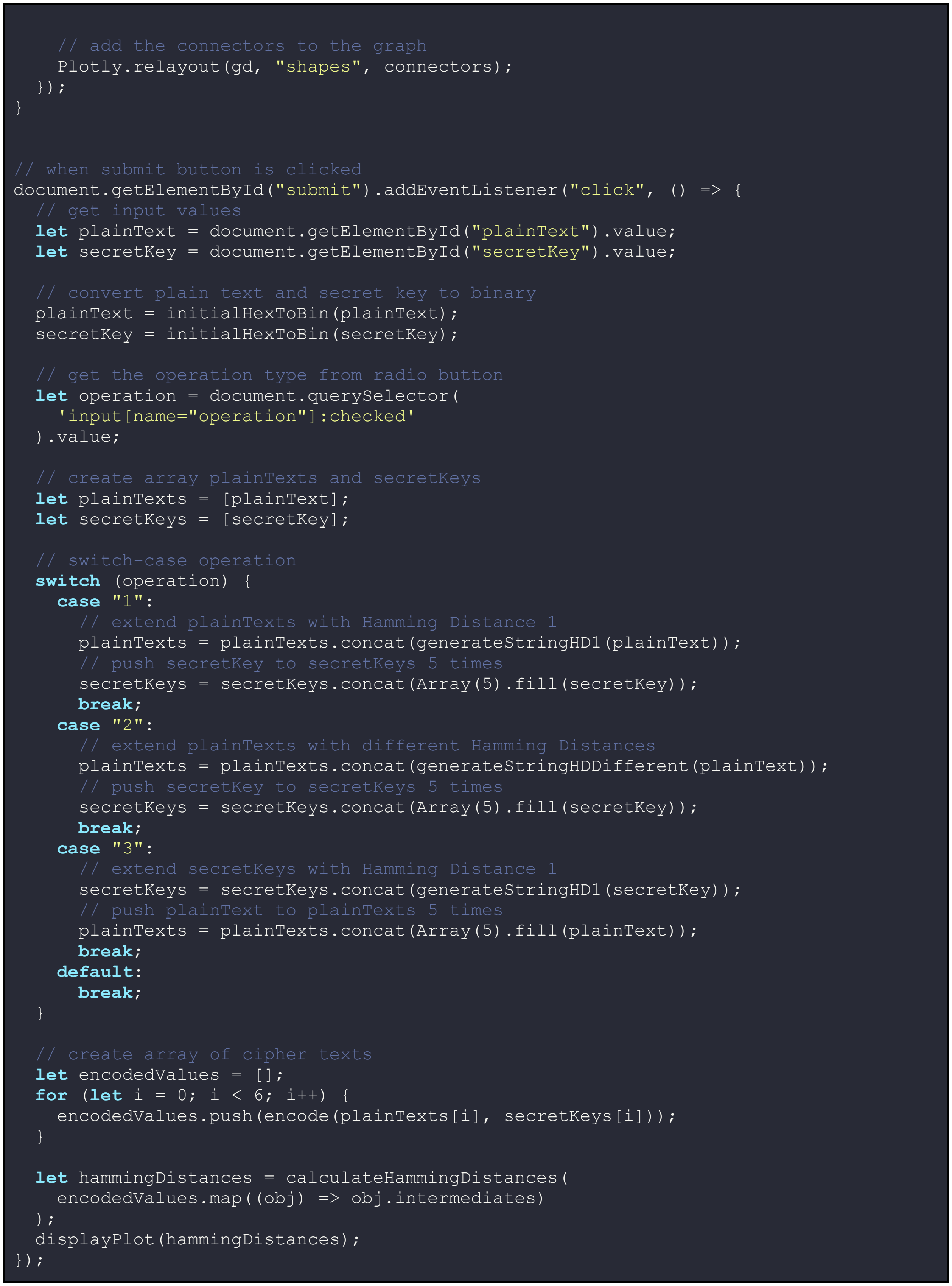
The JavaScript code involving logic (such as generating the plaintexts and ciphertexts, and plotting the graph) is shown below.

For complete source code, please refer to the repository link.

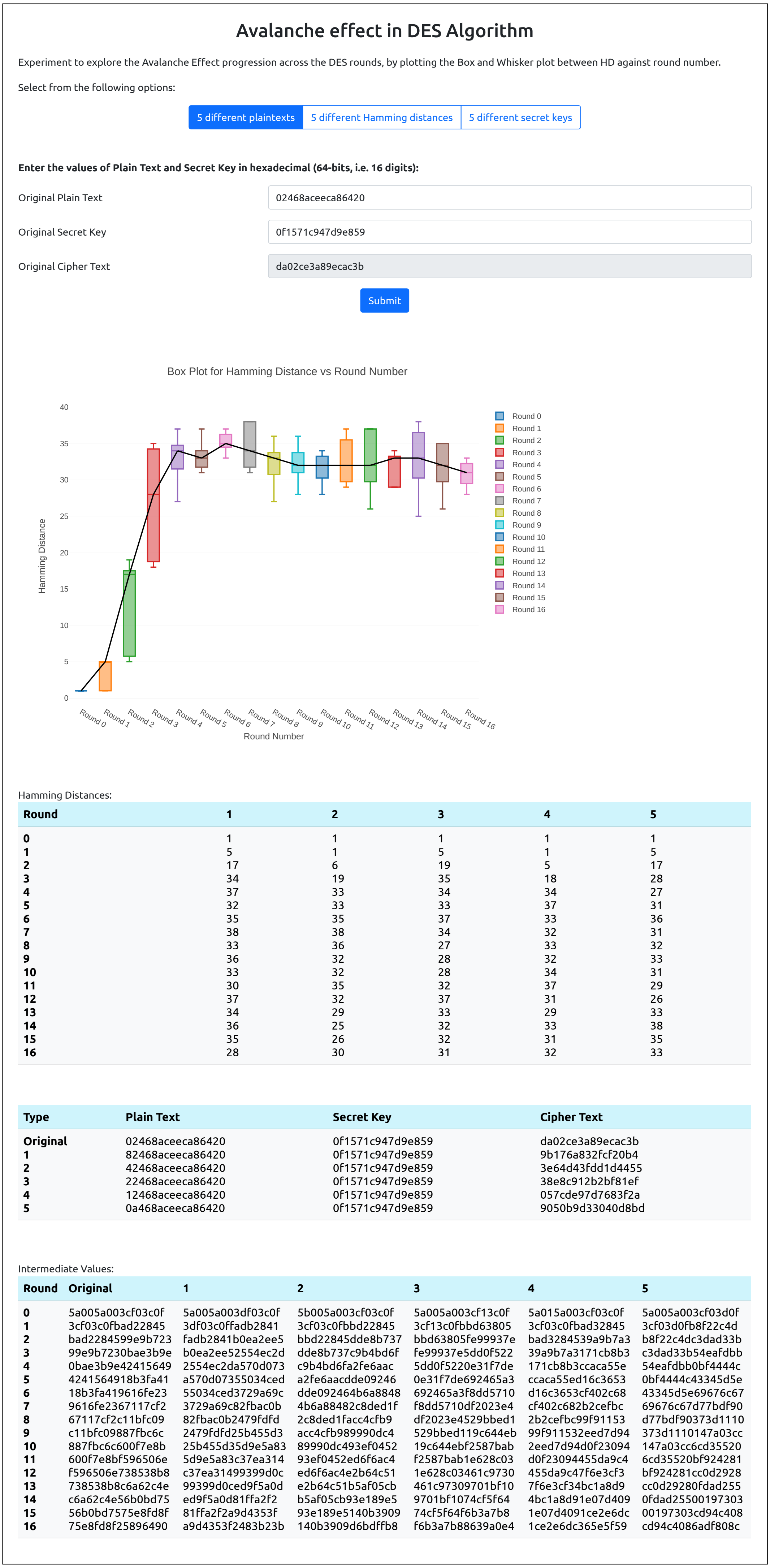
```
import { encode, initialHexToBin, finalBinToHex } from "./des.js";

// function to generate 5 strings with Hamming Distance 1, where str is binary
function generateStringHD1(str) {
  let diff = [0, 1, 2, 3, 4];
  let result = [];
  for (let i = 0; i < 5; i++) {
    let newStr = str;
    let index = diff[i];
    // flip the bit at index
    newStr = newStr.replaceAt(index, newStr[index] === "0" ? "1" : "0");
    result.push(newStr);
  }
  return result;
}

// function to generate 5 strings with different HD, where str is binary
function generateStringHDDifferent(str) {
  let hd = [[0], [0, 1], [0, 1, 2], [0, 1, 2, 3], [0, 1, 2, 3, 4]];
  let result = [];
  for (let i = 0; i < hd.length; i++) {
    // generate a string with hd[i] bits different from str
    let newStr = str;
    let bitsIndex = hd[i];
    for (let j = 0; j < bitsIndex.length; j++) {
      let index = bitsIndex[j];
      // flip the bit at index
      newStr = newStr.replaceAt(index, newStr[index] === "0" ? "1" : "0");
    }
    result.push(newStr);
  }
  return result;
}


// Calculate hamming distance between the intermediate values and original values
function calculateHammingDistances(intermediateValues) {
  let fullResult = [];
  // for all the 5 texts taken
  intermediateValues.forEach((values, position) => {
    let result = [];
    // for all 16 rounds
    values.forEach((value, r) => {
      let hd = 0;
      for (let i = 0; i < 64; i++) {
```



```
        // checking if bit at i is different from the 0th text, rth round, ith bit
        if (value[i] !== intermediateValues[0][r][i]) {
          hd++;
        }
      }
      result.push(hd);
    });
    fullResult.push(result);
  });
  return fullResult;
}

function displayPlot(hammingDistances) {
  let data = [];

  for (let round = 0; round <= 16; round++) {
    let result = [];
    // Pushing the hamming distance of each text, except the original text
    for (let i = 1; i <= 5; i++) {
      result.push(hammingDistances[i][round]);
    }
    data.push({
      y: result,
      type: "box",
      name: `Round ${round}`,
    });
  }

  let layout = {
    title: "Box Plot for Hamming Distance vs Round Number",
    xaxis: {
      title: "Round Number",
    },
    yaxis: {
      title: "Hamming Distance",
    },
    height: 650,
  };

  let config = {
    responsive: true,
  };

  Plotly.newPlot("plot", data, layout, config).then((gd) => {
    // calcdata is 2d
    // with length = # of traces on graph,
    var boxCalcData = gd.calcdata;

    var connectors = [];
    for (let i = 0; i < boxCalcData.length - 1; i++) {
      var box0 = boxCalcData[i][0];
      var box1 = boxCalcData[i + 1][0];

      // join the two adjacent boxes with a line
      connectors.push({
        type: "line",
        x0: box0.x,
        x1: box1.x,
        y0: box0.med,
        y1: box1.med,
      });
    }
```



```
    // add the connectors to the graph
    Plotly.relayout(gd, "shapes", connectors);
  });
}


// when submit button is clicked
document.getElementById("submit").addEventListener("click", () => {
  // get input values
  let plainText = document.getElementById("plainText").value;
  let secretKey = document.getElementById("secretKey").value;

  // convert plain text and secret key to binary
  plainText = initialHexToBin(plainText);
  secretKey = initialHexToBin(secretKey);

  // get the operation type from radio button
  let operation = document.querySelector(
    'input[name="operation"]:checked'
  ).value;

  // create array plainTexts and secretKeys
  let plainTexts = [plainText];
  let secretKeys = [secretKey];

  // switch-case operation
  switch (operation) {
    case "1":
      // extend plainTexts with Hamming Distance 1
      plainTexts = plainTexts.concat(generateStringHD1(plainText));
      // push secretKey to secretKeys 5 times
      secretKeys = secretKeys.concat(Array(5).fill(secretKey));
      break;
    case "2":
      // extend plainTexts with different Hamming Distances
      plainTexts = plainTexts.concat(generateStringHDDifferent(plainText));
      // push secretKey to secretKeys 5 times
      secretKeys = secretKeys.concat(Array(5).fill(secretKey));
      break;
    case "3":
      // extend secretKeys with Hamming Distance 1
      secretKeys = secretKeys.concat(generateStringHD1(secretKey));
      // push plainText to plainTexts 5 times
      plainTexts = plainTexts.concat(Array(5).fill(plainText));
      break;
    default:
      break;
  }

  // create array of cipher texts
  let encodedValues = [];
  for (let i = 0; i < 6; i++) {
    encodedValues.push(encode(plainTexts[i], secretKeys[i]));
  }

  let hammingDistances = calculateHammingDistances(
    encodedValues.map((obj) => obj.intermediates)
  );
  displayPlot(hammingDistances);
});
```



Avalanche effect in DES Algorithm

Experiment to explore the Avalanche Effect progression across the DES rounds, by plotting the Box and Whisker plot between HD against round number.

Select from the following options:

Enter the values of Plain Text and Secret Key in hexadecimal (64-bits, i.e. 16 digits):

Original Plain Text	02468aceeca86420
Original Secret Key	0f1571c947d9e859
Original Cipher Text	da02ce3a89ecac3b

Box Plot for Hamming Distance vs Round Number

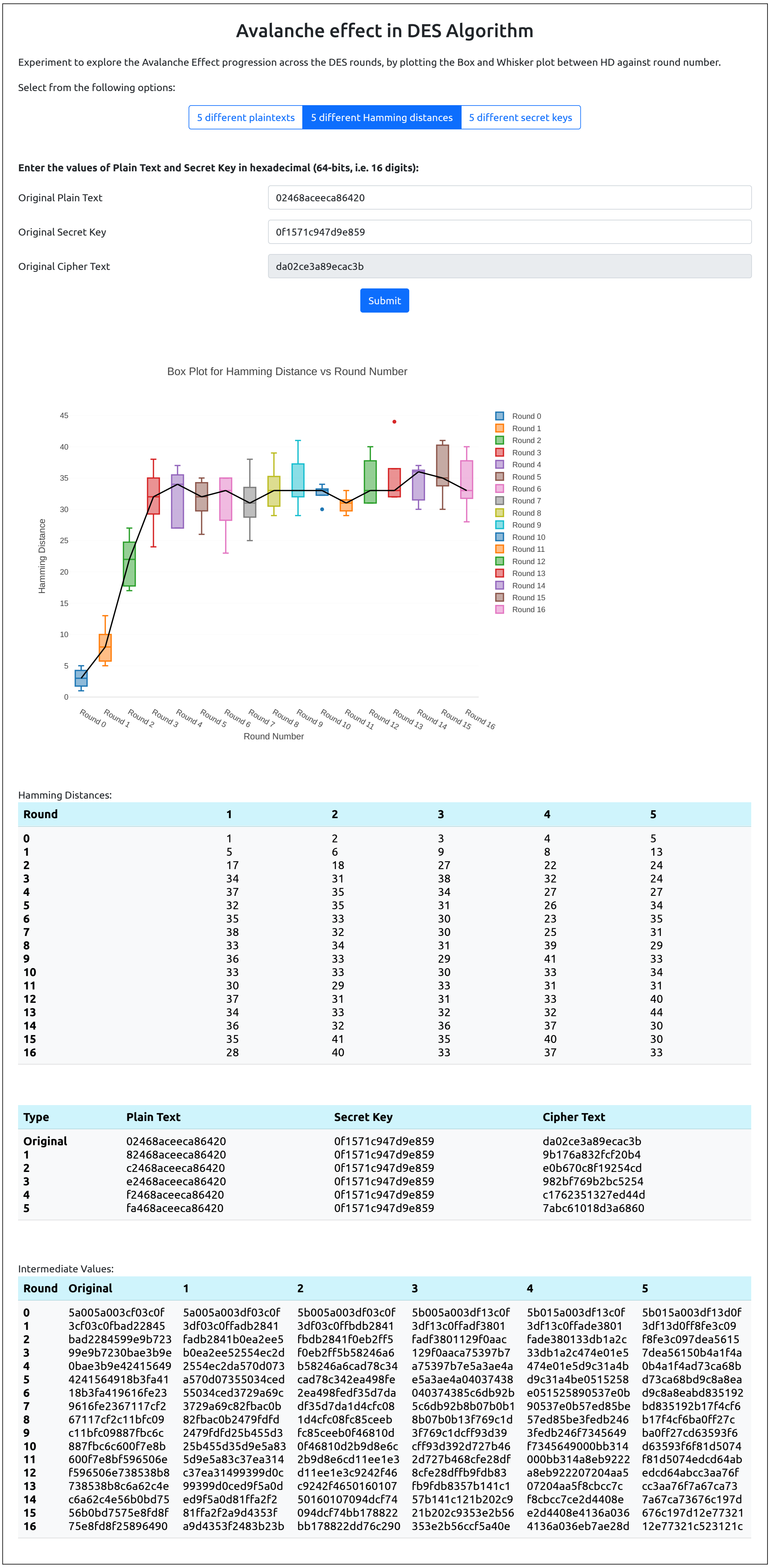
Hamming Distances:

Round	1	2	3	4	5
0	1	1	1	1	1
1	5	1	5	1	5
2	17	6	19	5	17
3	34	19	35	18	28
4	37	33	34	34	27
5	32	33	33	37	31
6	35	35	37	33	36
7	38	38	34	32	31
8	33	36	27	33	32
9	36	32	28	32	33
10	33	32	28	34	31
11	30	35	32	37	29
12	37	32	37	31	26
13	34	29	33	29	33
14	36	25	32	33	38
15	35	26	32	31	35
16	28	30	31	32	33

Type	Plain Text	Secret Key	Cipher Text
Original	02468aceeca86420	0f1571c947d9e859	da02ce3a89ecac3b
1	82468aceeca86420	0f1571c947d9e859	9b176a832fcf20b4
2	42468aceeca86420	0f1571c947d9e859	3e64d43fdd1d4455
3	22468aceeca86420	0f1571c947d9e859	38e8c912b2bf81ef
4	12468aceeca86420	0f1571c947d9e859	057cde97d7683f2a
5	0a468aceeca86420	0f1571c947d9e859	9050b9d33040d8bd

Intermediate Values:

Round	Original	1	2	3	4	5
0	5a005a003cf03c0f	5a005a003df03c0f	5b005a003cf03c0f	5a005a003cf13c0f	5a015a003cf03c0f	5a005a003cf03d0f
1	3cf03c0fbad22845	3df03c0ffadb2841	3cf03c0fbbd22845	3cf13c0fbbd63805	3cf03c0fbad32845	3cf03d0fb8f22c4d
2	bad2284599e9b723	fadb2841b0ea2ee5	bbd22845dde8b737	bbd63805fe99937e	bad3284539a9b7a3	b8f22c4dc3dad33b
3	99e9b7230bae3b9e	b0ea2ee52554ec2d	dde8b737c9b4bd6f	fe99937e5dd0f522	39a9b7a3171cb8b3	c3dad33b54eafdbb
4	0bae3b9e42415649	2554ec2da570d073	c9b4bd6fa2fe6aac	5dd0f5220e31f7de	171cb8b3ccaca55e	54eafdbb0bf4444c
5	4241564918b3fa41	a570d07355034ced	a2fe6aacdde09246	0e31f7de692465a3	ccaca55ed16c3653	0bf4444c43345d5e
6	18b3fa419616fe23	55034ced3729a69c	dde092464b6a8848	692465a3f8dd5710	d16c3653cf402c68	43345d5e69676c67
7	9616fe2367117cf2	3729a69c82fbac0b	4b6a88482c8ded1f	f8dd5710df2023e4	cf402c682b2cefbc	69676c67d77bdf90
8	67117cf2c11bfc09	82fbac0b2479fdfd	2c8ded1facc4cfb9	df2023e4529bbed1	2b2cefbc99f91153	d77bdf90373d1110
9	c11bfc09887fbc6c	2479fdfd25b455d3	acc4cfb989990dc4	529bbed119c644eb	99f911532eed7d94	373d1110147a03cc
10	887fbc6c600f7e8b	25b455d35d9e5a83	89990dc493ef0452	19c644ebf2587bab	2eed7d94d0f23094	147a03cc6cd35520
11	600f7e8bf596506e	5d9e5a83c37ea314	93ef0452ed6f6ac4	f2587bab1e628c03	d0f23094455da9c4	6cd35520bf924281
12	f596506e738538b8	c37ea31499399d0c	ed6f6ac4e2b64c51	1e628c03461c9730	455da9c47f6e3cf3	bf924281cc0d2928
13	738538b8c6a62c4e	99399d0ced9f5a0d	e2b64c51b5af05cb	461c97309701bf10	7f6e3cf34bc1a8d9	cc0d29280fdad255
14	c6a62c4e56b0bd75	ed9f5a0d81ffa2f2	b5af05cb93e189e5	9701bf1074cf5f64	4bc1a8d91e07d409	0fdad25500197303
15	56b0bd7575e8fd8f	81ffa2f2a9d4353f	93e189e5140b3909	74cf5f64f6b3a7b8	1e07d4091ce2e6dc	00197303cd94c408
16	75e8fd8f25896490	a9d4353f2483b23b	140b3909d6bdffb8	f6b3a7b88639a0e4	1ce2e6dc365e5f59	cd94c4086adf808c



Avalanche effect in DES Algorithm

Experiment to explore the Avalanche Effect progression across the DES rounds, by plotting the Box and Whisker plot between HD against round number.

Select from the following options:

Enter the values of Plain Text and Secret Key in hexadecimal (64-bits, i.e. 16 digits):

Original Plain Text: 02468aceeca86420

Original Secret Key: 0f1571c947d9e859

Original Cipher Text: da02ce3a89ecac3b

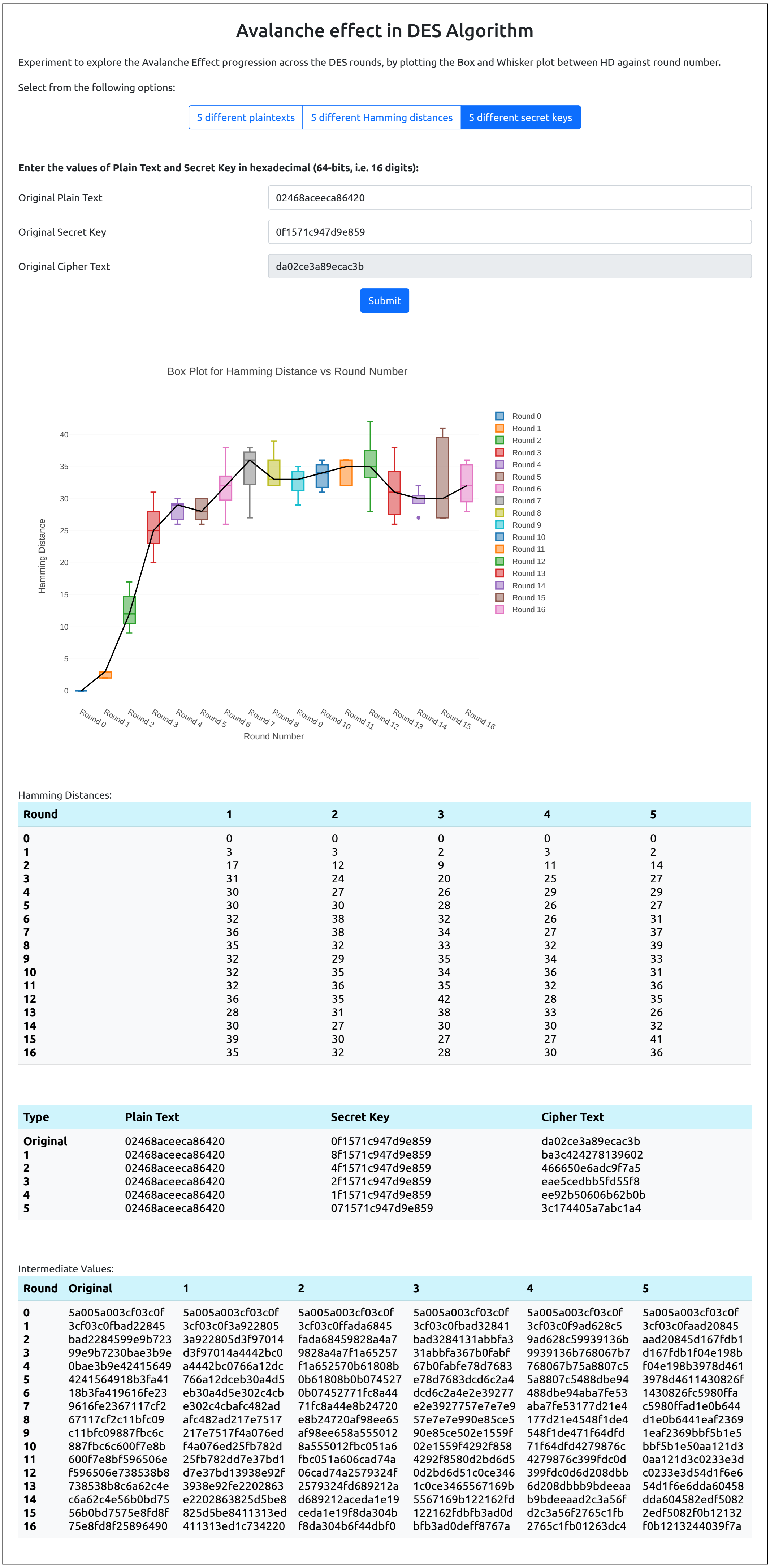
Hamming Distances:

Round	1	2	3	4	5
0	1	2	3	4	5
1	5	6	9	8	13
2	17	18	27	22	24
3	34	31	38	32	24
4	37	35	34	27	27
5	32	35	31	26	34
6	35	33	30	23	35
7	38	32	30	25	31
8	33	34	31	39	29
9	36	33	29	41	33
10	33	33	30	33	34
11	30	29	33	31	31
12	37	31	31	33	40
13	34	33	32	32	44
14	36	32	36	37	30
15	35	41	35	40	30
16	28	40	33	37	33

Type	Plain Text	Secret Key	Cipher Text
Original	02468aceeca86420	0f1571c947d9e859	da02ce3a89ecac3b
1	82468aceeca86420	0f1571c947d9e859	9b176a832fcf20b4
2	c2468aceeca86420	0f1571c947d9e859	e0b670c8f19254cd
3	e2468aceeca86420	0f1571c947d9e859	982bf769b2bc5254
4	f2468aceeca86420	0f1571c947d9e859	c1762351327ed44d
5	fa468aceeca86420	0f1571c947d9e859	7abc61018d3a6860

Intermediate Values:

Round	Original	1	2	3	4	5
0	5a005a003cf03c0f	5a005a003df03c0f	5b005a003df03c0f	5b005a003df13c0f	5b015a003df13c0f	5b015a003df13d0f
1	3cf03c0fbad22845	3df03c0ffadb2841	3df03c0ffbdb2841	3df13c0ffadf3801	3df13c0ffade3801	3df13d0ff8fe3c09
2	bad2284599e9b723	fadb2841b0ea2ee5	fbdb2841f0eb2ff5	fadf3801129f0aac	fade380133db1a2c	f8fe3c097dea5615
3	99e9b7230bae3b9e	b0ea2ee52554ec2d	f0eb2ff5b58246a6	129f0aaca75397b7	33db1a2c474e01e5	7dea56150b4a1f4a
4	0bae3b9e42415649	2554ec2da570d073	b58246a6cad78c34	a75397b7e5a3ae4a	474e01e5d9c31a4b	0b4a1f4ad73ca68b
5	4241564918b3fa41	a570d07355034ced	cad78c342ea498fe	e5a3ae4a04037438	d9c31a4be0515258	d73ca68bd9c8a8ea
6	18b3fa419616fe23	55034ced3729a69c	2ea498fedf35d7da	040374385c6db92b	e051525890537e0b	d9c8a8eabd835192
7	9616fe2367117cf2	3729a69c82fbac0b	df35d7da1d4cfc08	5c6db92b8b07b0b1	90537e0b57ed85be	bd835192b17f4cf6
8	67117cf2c11bfc09	82fbac0b2479fdfd	1d4cfc08fc85ceeb	8b07b0b13f769c1d	57ed85be3fedb246	b17f4cf6ba0ff27c
9	c11bfc09887fbc6c	2479fdfd25b455d3	fc85ceeb0f46810d	3f769c1dcff93d39	3fedb246f7345649	ba0ff27cd63593f6
10	887fbc6c600f7e8b	25b455d35d9e5a83	0f46810d2b9d8e6c	cff93d392d727b46	f7345649000bb314	d63593f6f81d5074
11	600f7e8bf596506e	5d9e5a83c37ea314	2b9d8e6cd11ee1e3	2d727b468cfe28df	000bb314a8eb9222	f81d5074edcd64ab
12	f596506e738538b8	c37ea31499399d0c	d11ee1e3c9242f46	8cfe28dffb9fdb83	a8eb922207204aa5	edcd64abcc3aa76f
13	738538b8c6a62c4e	99399d0ced9f5a0d	c9242f4650160107	fb9fdb8357b141c1	07204aa5f8cbcc7c	cc3aa76f7a67ca73
14	c6a62c4e56b0bd75	ed9f5a0d81ffa2f2	50160107094dcf74	57b141c121b202c9	f8cbcc7ce2d4408e	7a67ca73676c197d
15	56b0bd7575e8fd8f	81ffa2f2a9d4353f	094dcf74bb178822	21b202c9353e2b56	e2d4408e4136a036	676c197d12e77321
16	75e8fd8f25896490	a9d4353f2483b23b	bb178822dd76c290	353e2b56ccf5a40e	4136a036eb7ae28d	12e77321c523121c



Avalanche effect in DES Algorithm

Experiment to explore the Avalanche Effect progression across the DES rounds, by plotting the Box and Whisker plot between HD against round number.

Select from the following options:

Enter the values of Plain Text and Secret Key in hexadecimal (64-bits, i.e. 16 digits):

Original Plain Text: 02468aceeca86420

Original Secret Key: 0f1571c947d9e859

Original Cipher Text: da02ce3a89ecac3b

Box Plot for Hamming Distance vs Round Number

Hamming Distances:

Round	1	2	3	4	5
0	0	0	0	0	0
1	3	3	2	3	2
2	17	12	9	11	14
3	31	24	20	25	27
4	30	27	26	29	29
5	30	30	28	26	27
6	32	38	32	26	31
7	36	38	34	27	37
8	35	32	33	32	39
9	32	29	35	34	33
10	32	35	34	36	31
11	32	36	35	32	36
12	36	35	42	28	35
13	28	31	38	33	26
14	30	27	30	30	32
15	39	30	27	27	41
16	35	32	28	30	36

Type	Plain Text	Secret Key	Cipher Text
Original	02468aceeca86420	0f1571c947d9e859	da02ce3a89ecac3b
1	02468aceeca86420	8f1571c947d9e859	ba3c424278139602
2	02468aceeca86420	4f1571c947d9e859	466650e6adc9f7a5
3	02468aceeca86420	2f1571c947d9e859	eae5cedbb5fd55f8
4	02468aceeca86420	1f1571c947d9e859	ee92b50606b62b0b
5	02468aceeca86420	071571c947d9e859	3c174405a7abc1a4

Intermediate Values:

Round	Original	1	2	3	4	5
0	5a005a003cf03c0f	5a005a003cf03c0f	5a005a003cf03c0f	5a005a003cf03c0f	5a005a003cf03c0f	5a005a003cf03c0f
1	3cf03c0fbad22845	3cf03c0f3a922805	3cf03c0ffada6845	3cf03c0fbad32841	3cf03c0f9ad628c5	3cf03c0faad20845
2	bad2284599e9b723	3a922805d3f97014	fada68459828a4a7	bad3284131abbfa3	9ad628c59939136b	aad20845d167fdb1
3	99e9b7230bae3b9e	d3f97014a4442bc0	9828a4a7f1a65257	31abbfa367b0fabf	9939136b768067b7	d167fdb1f04e198b
4	0bae3b9e42415649	a4442bc0766a12dc	f1a652570b61808b	67b0fabfe78d7683	768067b75a8807c5	f04e198b3978d461
5	4241564918b3fa41	766a12dceb30a4d5	0b61808b0b074527	e78d7683dcd6c2a4	5a8807c5488dbe94	3978d4611430826f
6	18b3fa419616fe23	eb30a4d5e302c4cb	0b07452771fc8a44	dcd6c2a4e2e39277	488dbe94aba7fe53	1430826fc5980ffa
7	9616fe2367117cf2	e302c4cbafc482ad	71fc8a44e8b24720	e2e3927757e7e7e9	aba7fe53177d21e4	c5980ffad1e0b644
8	67117cf2c11bfc09	afc482ad217e7517	e8b24720af98ee65	57e7e7e990e85ce5	177d21e4548f1de4	d1e0b6441eaf2369
9	c11bfc09887fbc6c	217e7517f4a076ed	af98ee658a555012	90e85ce502e1559f	548f1de471f64dfd	1eaf2369bbf5b1e5
10	887fbc6c600f7e8b	f4a076ed25fb782d	8a555012fbc051a6	02e1559f4292f858	71f64dfd4279876c	bbf5b1e50aa121d3
11	600f7e8bf596506e	25fb782dd7e37bd1	fbc051a606cad74a	4292f8580d2bd6d5	4279876c399fdc0d	0aa121d3c0233e3d
12	f596506e738538b8	d7e37bd13938e92f	06cad74a2579324f	0d2bd6d51c0ce346	399fdc0d6d208dbb	c0233e3d54d1f6e6
13	738538b8c6a62c4e	3938e92fe2202863	2579324fd689212a	1c0ce3465567169b	6d208dbbb9bdeeaa	54d1f6e6dda60458
14	c6a62c4e56b0bd75	e2202863825d5be8	d689212aceda1e19	5567169b122162fd	b9bdeeaad2c3a56f	dda604582edf5082
15	56b0bd7575e8fd8f	825d5be8411313ed	ceda1e19f8da304b	122162fdbfb3ad0d	d2c3a56f2765c1fb	2edf5082f0b12132
16	75e8fd8f25896490	411313ed1c734220	f8da304b6f44dbf0	bfb3ad0deff8767a	2765c1fb01263dc4	f0b1213244039f7a