CHAPTER 3
RESEARCH METHODOLOGY

3.1. Data Encryption Standard (DES)

Plaintext of DES can be encrypted in a group of 8 digit characters or 16 hexadecimal numbers or 64 bits. The key for input DES also needs 8 characters or 16 hexadecimal numbers or 64 bits long. But only 56 bits are used because every 8th key bit will be ignored.

Firstly, convert the plaintext and the key into binary bits. Because DES is using block cipher method, the plaintext and the key are divided into a block of 8 bytes length. The second step, permute the 64-bit key using PC-1 table. The first bit of 56-bit permutation key will be the 57th bit of the original key and so on. Only 56 bits as the result of permutation key.

Table 3.1: PC-1
(source : http://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm)

PC-1						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Then, we make 16 blocks by dividing the key into two parts, C_0 for left and D_0 for right. Each pair of Cn and Dn blocks will be obtained from the previous pair, namely Cn-1 and Dn-1, for n between 1 and 16, using the left shift. Move each bit to the left one place, for the first bit move it to the end of the block. The scheme as below,

Table 3.2: Left Shift
(source : http://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm)

Iteration Number	Number of Left Shifts
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

Now applying the second permutation table to each of the 16 ***CnDn*** pair keys. **PC-2** only uses 48 of these.

Table 3.3: PC-2
(source : http://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm)

PC-2					
14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

This is the end of our encryption keys. Now encode the message.

First, apply the initial permutation (IP) to each block of 64 bits plaintext, the table of IP is on below.

Table 3.4: IP
(source : http://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm)

IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

After that, we divide the permuted block to left as L_0 (32 bits) and right as R_0 (32 bits). Then 16 iterations are performed, for 1 <= n <= 16, using the function f which operates on 2 blocks (32-bit data blocks and Kn 48-bit keys) to produce 32-bit blocks. The + symbol indicates the addition of XOR. For **n** from 1-16 we compute:

$$\mathbf{Ln = Rn\text{-}1}$$

$$\mathbf{Rn = Ln\text{-}1 + f(Rn\text{-}1,Kn)}$$

The final block, for n = 16 is $L_{16}R_{16}$. In each iteration, we take the 32 bits right from the previous result and make the 32 bits left from the current step. In the current step for the right 32 bits, we XOR 32 bits left from the previous step with calculation f. Next, use E bit-selection table to expand from 32 bits to 48 bits each block of $\boldsymbol{R_{n-1}}$.

Table 3.5: E Bit-Selection Table
(source : http://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm)

E BIT-SELECTION TABLE					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

After that, XOR the output $\mathbf{E}(\boldsymbol{R}_{n-1})$ with the key $\boldsymbol{K}_n$:

$$\boldsymbol{K}_n + \mathbf{E}(\boldsymbol{R}_{n-1}).$$

Now we have 48 bits, to addresses in **S boxes** tables. There are 8 S boxes for every 6 bits. The original 6 bits will be replaced by 4-bit number. The final result is 8 groups of 4 bits.

The tables $\boldsymbol{S}_1,\ldots,\boldsymbol{S}_8$ are below:

Table 3.6: S-Boxes
(source : http://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm)

S1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

The last step in the calculation f is to do the permutation of the output of the S-box to get the final value f:

$$f = \mathbf{P}(S_1(B_1)S_2(B_2)...S_8(B_8))$$

Permutation P is defined in the following table. P produces a 32-bit output from 32-bit input by permuting the input block bits.

Table 3.7: P
(source : http://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm)

P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Next round, get the $L_2 = R_1$ and $R_2 = L_1 + f(R_1, K_2)$ and so on for 16 rounds. At the end of the sixteenth round, we have blocks L_{16} and R_{16}. Then we ***reverse*** two blocks to the 64-bit block.

$$\mathbf{R_{16}L_{16}}$$

And apply a final permutation $\mathbf{IP^{-1}}$ as defined by the following table:

Table 3.8: IP^{-1}
(source : http://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm)

IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

The result of permutation IP^{-1} is the ciphertext in bit format. We covert it into hexadecimal in Arduino code. Descriptions are just the opposite of encryption, following the same steps as above, but reversing the order in which the subkeys are applied.

There are many block cipher mode of operations. If in DES, each 64-bit block is encrypted individually, then the mode is called Electronic Code Book (ECB). Other modes of DES encryption are Chain Block Coding (CBC) and Cipher Feedback (CFB), which makes each block cipher depend on all previous message blocks through the initial XOR operation.

3.2. Advanced Encryption Standard (AES)

AES also was known as Rijndael, using block cipher system too. There are 3 types of AES that have different key length, which is AES-128, AES-192, and AES-256. The difference between the 3 types is the number of rounds. AES-128 do 10 rounds, AES-192 do 12 rounds, and AES-256 do 14 rounds. This project use AES-128 where the 128 bits of a plaintext block as 16 bytes (16 digits data) will be arranged in 4 columns and 4 rows to be processed in a matrix.

First, the 128-bit key is converted into hexadecimal then entered into the 4x4 size matrix which called as state array. 16 digits of plaintext convert into hexadecimal and also entered into the 4x4 size matrix. There will be 9 rounds with four types of operations (add round key, sub bytes, shift rows, mix columns). The tenth round does a slightly different operation than the others because it didn’t do mix columns. For the first round can be seen below:

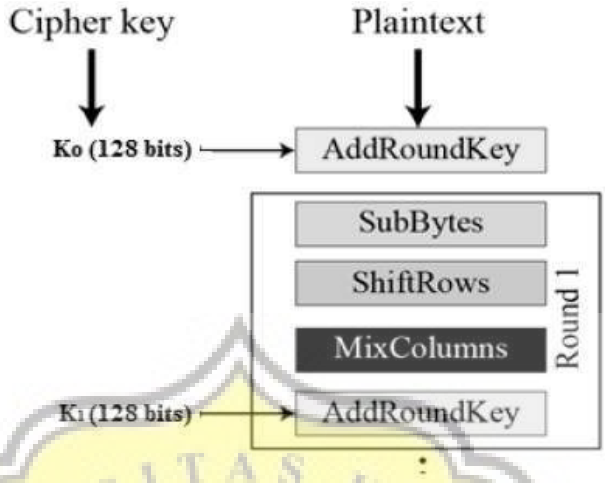

Illustration 3.1: Round 1
(source : https://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm)

1. Add Round Key

   Add Round Key is basically combining existing ciphertext with cipher key with XOR relationship.

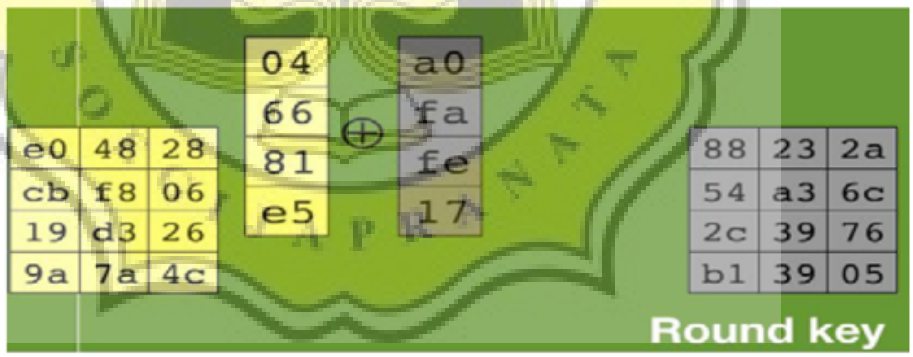

Illustration 3.2: Add Round Key
(source : http://klinikinformatikacyber.blogspot.com/2016/03/pengertian-dan-sistem-kerja-aes-iii.html)

In the image on the left is the ciphertext and the right is the round key. XOR is done per column, namely column-1 text cipher in XOR with column-1 round key and so on.

2. Sub Bytes

The principle of Sub Bytes is to swap the contents of the existing matrix/table with another matrix/table called Rijndael S-Box. Below are examples of Sub Bytes and Rijndael S-Box.

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Illustration 3.3: Rijndael S-Box
(source : http://klinikinformatikacyber.blogspot.com/2016/03/pengertian-dan-sistem-kerja-aes-iii.html)

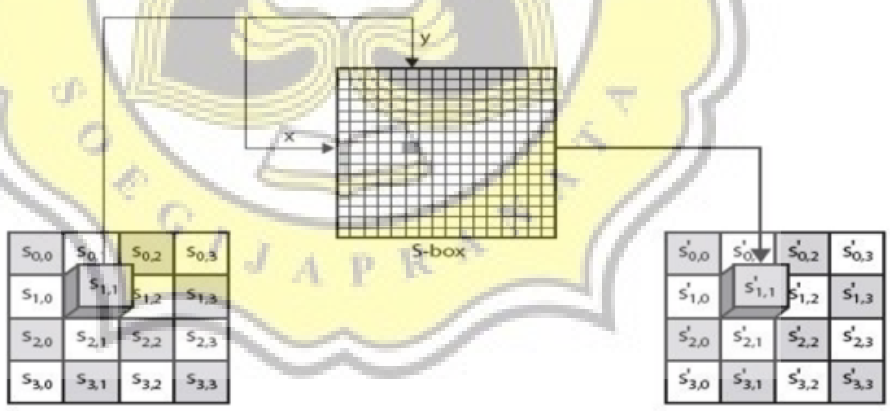

Illustration 3.4: Sub Bytes Illustration
(source: http://klinikinformatikacyber.blogspot.com/2016/03/pengertian-dan-sistem-kerja-aes-iii.html)

In the illustration of Sub Bytes above, there are column numbers and row numbers. As mentioned earlier, each box contents of the cipher block contain hexadecimal information consisting of two digits, can be numbers, numbers, or letters all listed in Rijndael S-Box. The step is to take one of the contents of the matrix box, matching it with the left

digit as the row and right digit as the column. Then by knowing the columns and rows, we can retrieve a table from Rijndael S-Box. The final step is to change the entire block of cipher into a new block whose contents are the result of exchanging all the contents of the block with the contents of the steps mentioned earlier.

3. Shift Rows

Shift Rows as the name suggests is a process that does shift on each element of the block/table that is done per line. The first line is not shifted, the second line is shifted by 1 byte, the third row is shifted by 2 bytes, and the fourth row is shifted by 3 bytes. The shift seen in a block is a shift of each element to the left depending on how many bytes it displaces, each shift of 1 byte means shifting to the left one time.

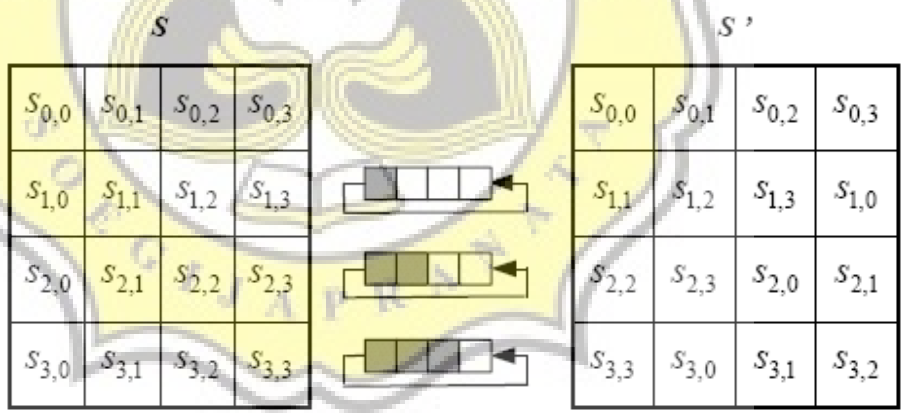

Illustration 3.5: Shift Rows
(source : https://www.researchgate.net/figure/AES-ShiftRows-function-taken-from-8_fig4_265112905)

4. Mix Columns

What happens when Mix Column is multiplying each element of the cipher block with the matrix shown in Table 3.9. Multiplication is done like ordinary matrix multiplication, using a dot product and multiplying both of them into a new cipher block. The illustration in

Illustration 3.6 will explain how this multiplication should be done. In this way, the whole process that occurs in AES has been explained.

Table 3.9: Mix Columns
(source: http://klinikinformatikacyber.blogspot.com/2016/03/pengertian-dan-sistem-kerja-aes-iii.html)

02	01	01	03
03	02	01	01
01	03	02	01
01	01	02	03

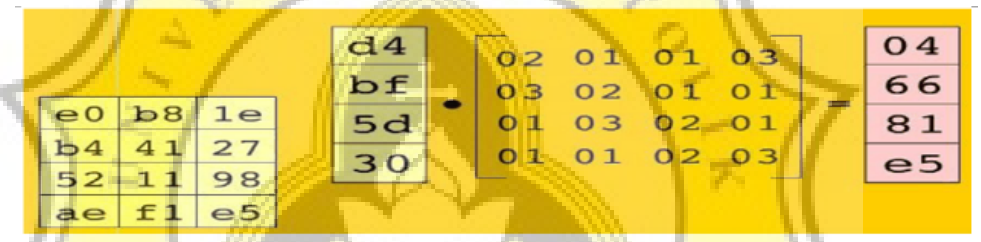

Illustration 3.6: Mix Columns Illustration
(source : http://daffasempurna.blogspot.com/2016/06/pengertian-dan-sistem-kerja-advanced.html)

5. AES Flow Diagram

All processes described previously can be seen in the following picture. Where starting from the second round, continuous repetition is carried out with a series of processes of Sub Bytes, Shift Rows, Mix Columns, and Add Round Key, after which the round results will be used in the next round with the same method, but in the tenth round, Mix Columns process was skipped. In other words, the process sequence that was carried out were Sub Bytes, Shift Rows, and Add Round Key, the results of the Add Round Key were used as the ciphertext of AES.

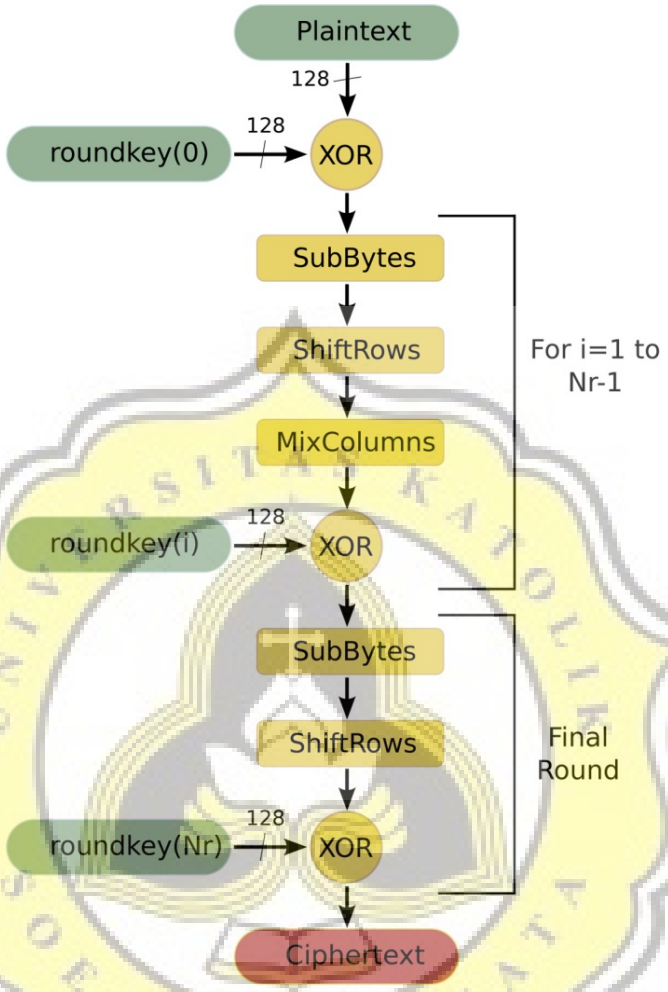

Illustration 3.7: AES Flowchart
(source : https://www.researchgate.net/figure/Flowchart-of-the-AES-algorithm-Encryption-process_fig4_233828516)

3.3. DES and AES Implementations on Arduino

The first thing to do in this research is to implement DES and AES algorithms on Arduino with C language. A variable that stores text (in the form of a String) originating from a sensor, is encrypted using a specified key. The encryption results are ciphertext sent to the PHP server. The ciphertext is decrypted with the same key using DES and AES on php. The results are displayed and matched whether they are in accordance with the one sent.

3.4. DES and AES Implementations on PHP

The decryption of data/ciphertext from Arduino is done using php. The ciphertext in the form of string is decrypted then matched with the plaintext when it has not been encrypted.

3.5. Installation of Sensors

As a simulation, the sensors used in this study are LDR and Potentiometer. In the same program with DES / AES code, the sensor results are read and encrypted.

3.6. Compare the DES and AES Encryption Speeds

The speed of data encryption in DES and AES is done on the same program using the micro (ms) time unit. The research takes 2 kinds of encryption time, there are encryption time per piece and total encryption time.

3.7. Compare the Power Consumption to Process DES and AES

To find out the power consumption in Arduino when executing DES / AES is by using the ACS712 sensor. The ACS712 sensor connected to the adapter and the Arduino that execute DES/AES, then calculates the power during the program.

3.8. Testing

After all sensors and components are installed, testing of the tool is carried out. Starting from data collection from sensors, data encryption process, sending to the server, until finally decrypted and displayed the results of the sensor. The time data used for DES / AES encryption and the power consumption are recorded for comparison, then the results of the research are concluded.