CHAPTER 4

ANALYSIS AND DESIGN

4.1 Analysis

4.1.1 Base64

The Base64 algorithm uses plain text with the size of each character 1 byte or 8 bits. In encryption the data will be encoded and decoded into ASCII format, which is based on 64 base numbers. Then the binary will be split into 6 bits, after that it will be converted back to decimal. After it becomes a decimal form, the decimal is changed to Base64 table. The results of the encryption on the Base64 transformation consist of A..Z, a..z and 0..9, and are added with the last two symbols which are + and / and one character equal to (=) which is used to adjust and fulfill binary data or pad fillers. In the decryption the reverse algorithm is performed.

4.1.2 Least Significant Bit (LSB)

The Least Significant Bit algorithm uses plain text with the size of each character 1 byte or 8 bits. In encryption into the WAV audio file the input text in the form of a string is made into a byte array. Then the byte array will be split into bits. This bit will replace the final bit of the WAV audio file. The final process of WAV audio files will be reconstructed again. In the decryption the reverse algorithm is performed.

4.1.3 Encryption and Decryption Speed

Encryption and decryption speed measurements using ms units. Encryption time is calculated from the time from encryption to encryption. Decryption time is also calculated from the time from decryption to decryption.

4.2 Desain

4.2.1 Encryption

Encryption in this project uses a combination of the Base64 algorithm and the Least Significant Bit (LSB). Encryption starts from the Base64 algorithm. Here's how the Base64 algorithm works as an example of encrypting text / messages with the word "UNIKA". The first step is to find the decimal of each character (decimal is searched through ASCII table) then convert to binary numbers.

Decimal "U" character: 85

85/2 there is a residual quotient 1

42/2 there is a residual quotient 0

21/2 there is a residual quotient 1

10/2 there is a residual quotient 0

5/2 there is a residual quotient 1

2/2 there is a residual quotient 0

1/2 there is a residual quotient 1

0/2 there is a residual quotient 0

The results are sorted from the bottom up, so the binary "U" character is 01010101

Decimal "N" character: 78

78/2 there is a residual quotient 0

39/2 there is a residual quotient 1

19/2 there is a residual quotient 1

9/2 there is a residual quotient 1

4/2 there is a residual quotient 0

2/2 there is a residual quotient 0

1/2 there is a residual quotient 1

0/2 there is a residual quotient 0

The results are sorted from the bottom up, so the binary "N" character is 01001110

Decimal "I" character: 73

73/2 there is a residual quotient 1

36/2 there is a residual quotient 0

18/2 there is a residual quotient 0

9/2 there is a residual quotient 1

4/2 there is a residual quotient 0

2/2 there is a residual quotient 0

1/2 there is a residual quotient 1

0/2 there is a residual quotient 0

The results are sorted from the bottom up, so the binary character "I" is 01001001

Decimal "K" character: 75

75/2 there is a residual quotient 1

37/2 there is a residual quotient 1

18/2 there is a residual quotient 0

9/2 there is a residual quotient 1

4/2 there is a residual quotient 0

2/2 there is a residual quotient 0

1/2 there is a residual quotient 1

0/2 there is a residual quotient 0

The results are sorted from the bottom up, so the binary "K" character is 01001011

Decimal "A" character: 65

65/2 there is a residual quotient 1

32/2 there is a residual quotient 0

16/2 there is a residual quotient 0

8/2 there is a residual quotient 0

4/2 there is a residual quotient 0

2/2 there is a residual quotient 0

1/2 there is a residual quotient 1

0/2 there is a residual quotient 0

The results are sorted from the bottom up, so the binary character "A" is 01000001

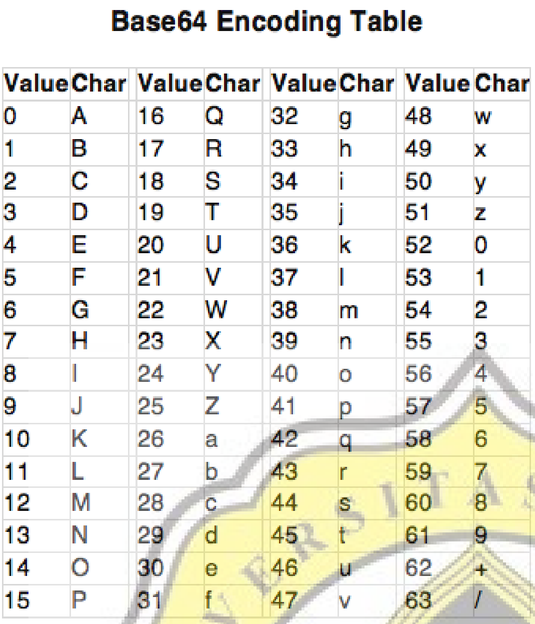
The second step combines all the binaries of the word "UNIKA". If combined it will become 0101010101001110010010010100101101000001. The binary will then be split into 6 bits into 010101 010100 111001 001001 010010 110100 0001. In the last part there are 4 binary bits which will be changed to 6 bits to 000100. The final result after being broken down into 6 bits is 010101 010100 111001 001001 010010 110100 0001. In the last part there are 4 binary bits which will be changed to 6 bits to 000100. 111001 001001 010010 110100 000100. After the binary becomes 6 bits, the 6 bit binary will be converted back to decimal. Example steps for binary to decimal conversion :

010101 = $(0*2^5) + (1*2^4) + (0*2^3) + (1*2^2) + (0*2^1) + (1*2^0)$

010101 = 0 + 16 + 0 + 4 + 0 + 1 = 21

So the decimal of the binary is 21.

After all the 6 bit binaries are converted to decimal will get a result of 21,20,57,9,18,52,4. This number will be converted again into Base64 table.



Base64 Encoding Table

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

21 = V

20 = U

57 = 5

9 = J

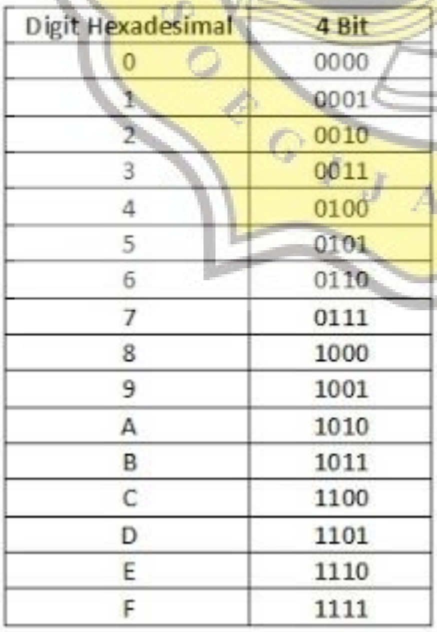
18 = S

52 = O

4 = E

The results of encryption into Base64 from the word "UNIKA" to VU5JSOE

After getting the results of Base64 encryption, VU5JSOE. The VU5JSOE string will be converted to a byte array. The process of converting it to a byte array, which is the V5SJOE character, will be found hexadecimal from the ASCII table. The resulting byte array is = {0x56,0x35,0x53,0x4A, 0x4F, 0x45,0x00} (0x00 = null). Then the byte array will be converted into binary form. How to convert a byte array to binary using a binary table, for example 0x56:



Digit Hexadesimal	4 Bit
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

5 = 0101

6 = 0110

The binary becomes 0b01010110 (0b is only the code for hexadecimal numbers)

So the final result of conversion into binary form for byte array {0x56, 0x35, 0x53, 0x4A, 0x4F, 0x45, 0x00} is 0b01010110, 0b00110101, 0b01010011, 0b01001010, 0b01001111, 0b01000101, 0b0000. Then the binary will be broken down into bits. The step is to break the binary into bits, for example binary 0b01010110

0b0000000**0**

0b0000000**1**

0b0000000**0**

0b0000000**1**

0b0000000**0**

0b0000000**1**

0b0000000**1**

0b0000000**0**

The final result after the binary VU5JSOE is broken down into bits :

0b0000000**0**	0b0000000**0**	0b0000000**0**	0b0000000**0**	0b0000000**0**	0b0000000**0**	0b0000000**0**
0b0000000**1**	0b0000000**0**	0b0000000**1**	0b0000000**1**	0b0000000**1**	0b0000000**1**	0b0000000**0**
0b0000000**0**	0b0000000**1**	0b0000000**0**	0b0000000**0**	0b0000000**0**	0b0000000**0**	0b0000000**0**
0b0000000**1**	0b0000000**1**	0b0000000**1**	0b0000000**0**	0b0000000**0**	0b0000000**0**	0b0000000**0**
0b0000000**0**	0b0000000**0**	0b0000000**0**	0b0000000**1**	0b0000000**1**	0b0000000**0**	0b0000000**0**
0b0000000**1**	0b0000000**1**	0b0000000**0**	0b0000000**0**	0b0000000**1**	0b0000000**1**	0b0000000**0**
0b0000000**1**	0b0000000**0**	0b0000000**1**	0b0000000**1**	0b0000000**1**	0b0000000**0**	0b0000000**0**
0b0000000**0**	0b0000000**1**	0b0000000**1**	0b0000000**0**	0b0000000**1**	0b0000000**1**	0b0000000**0**

After getting the fraction bits from the VU5JSOE character, the fraction will replace the last bit of the audio file. Example of a WAV audio file bit :

0b1110100001110010	0b.........1	0b.........1	0b.........0	0b.........0	0b.........1	0b.........1
0b1110110001110011	0b.........0	0b.........1	0b.........1	0b.........1	0b.........0	0b.........1

0b1110100101110011	0b.........1	0b.........1	0b.........1	0b.........0	0b.........0	0b.........1
0b1110101001110011	0b.........0	0b.........1	0b.........1	0b.........1	0b.........1	0b.........1
0b1110111101110010	0b.........1	0b.........1	0b.........0	0b.........0	0b.........0	0b.........1
0b1110100000010011	0b.........0	0b.........1	0b.........1	0b.........0	0b.........0	0b.........0
0b1110101001110010	0b.........0	0b.........1	0b.........0	0b.........0	0b.........0	0b.........0
0b1100100001010011	0b.........1	0b.........0	0b.........1	0b.........0	0b.........1	0b.........0

The last bit of the WAV audio file is then changed to 0 :

0b111010000111001**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**
0b111011000111001**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**
0b111010010111001**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**
0b111010100111001**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**
0b111011110111001**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**
0b111010000001001**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**
0b111010100111001**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**
0b110010000101001**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**

Then the last bit in the WAV audio file after 0 will be replaced by the VU5JSOE in-law bit :

0b111010000111001**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**
0b111011000111001**1**	0b.........**0**	0b.........**1**	0b.........**1**	0b.........**1**	0b.........**1**	0b.........**0**
0b111010010111001**0**	0b.........**1**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**
0b111010100111001**1**	0b.........**1**	0b.........**1**	0b.........**0**	0b.........**0**	0b.........**0**	0b.........**0**
0b111011110111001**0**	0b.........**0**	0b.........**0**	0b.........**1**	0b.........**1**	0b.........**0**	0b.........**0**
0b111010000001001**1**	0b.........**1**	0b.........**0**	0b.........**0**	0b.........**1**	0b.........**1**	0b.........**0**
0b111010100111001**1**	0b.........**0**	0b.........**1**	0b.........**1**	0b.........**1**	0b.........**0**	0b.........**0**

0b110010000101001**0**	0b.........**1**	0b.........**1**	0b.........**0**	0b.........**1**	0b.........**1**	0b.........**0**

After LSB is applied to the WAV audio file, the WAV audio file will be reconstructed again to produce a new WAV audio file that has been inserted by the message.

4.2.2 Decryption

For the decryption method, the only steps to reverse are the same steps as above by reversing the algorithm. To illustrate the decryption process, following the decryption algorithm flowchart.

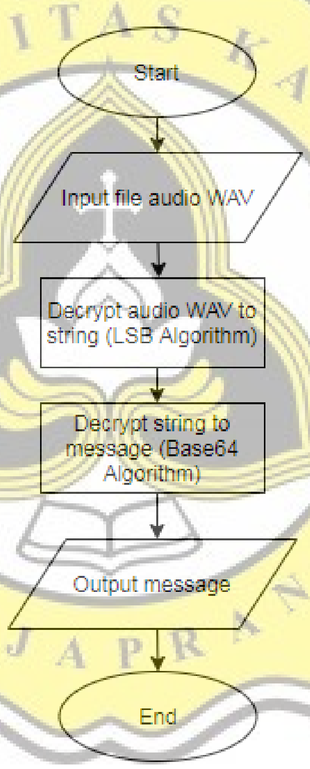

Start
Input file audio WAV
Decrypt audio WAV to string (LSB Algorithm)
Decrypt string to message (Base64 Algorithm)
Output message
End

Illustration 4.1: Decryption Flowchart