LIGHT WEIGHT AND PRIVACY SECURING MEDICAL SERVICES ACCESS FOR HEALTHCARE CLOUD

Ass.t Professor Shashikala H.K, Shaik Sharuk Ahmed, Pidugu Rajasekhar Reddy

Computer Science and Engineering Department, Jain University

Ramanagara District,Karnataka 562112,India

Abstract— **With the rapid climb of cloud computing technology, healthcare cloud system is turning into progressively fashionably, that reduces the time of unwellness diagnosing and brings nice convenience to people's lives, meantime the healthcare cloud system sometimes involves user's non-public information, and there is still a challenge the way to make sure that the sensitive information of users is not disclosed. Advanced encryption standard (AES) could be a terribly helpful technique for the privacy protection of users and is incredibly appropriate for anonymous authentication and privacy access management.**

I. INTRODUCTION

Recently, the event of cloud computing has brought new breakthroughs in many fields, such as healthcare, transportation, education, finance, and energy. however ancient healthcare system cannot meet the industrial healthcare services for its certain in efficiencies. With the advantage of cloud computing, the patient centered medical information system can realize the sharing of medical resources. In such a system, healthcare services of various medical institutions are deployed in healthcare cloud.

This arises the matter of data security. To unravel this drawback a data is stored or transmitted in the encrypted format. This encrypted data is illegible to the unauthorized user. Cryptography could be a science of information security that secures the data while the data is being transmitted and stored.

Every encryption and decryption methods have 2 aspects: The algorithm and the key use for the encryption and decryption. However, it's the key used for encryption and decryption that creates the method of cryptography secure. There are two types of cryptographic mechanisms: symmetrical key cryptography within which the concept key is used for encryption and decryption. just in case of uneven key cryptography 2 different keys are used for encryption and decryption. Symmetric key algorithm is a way quicker and easier to implement and needed less process power as compare to asymmetric key algorithm.

2. AES ALGORITHM SPECIFICATION

AES algorithm is of 3 types i.e. AES-128, AES-192 and AES-256. This classification is completed on the bases of the key used in the algorithm for encryption and decryption process. The numbers represent the dimensions of key in bits. This key size determines the protection level because the size of key will increase the extent of security will increases. The AES algorithm uses a spherical function that's composed of 4 different byte-oriented transformations. For encryption purpose four rounds consist of:

- Substitute byte
- Shift row
- Mix columns
- Add round key

While the decryption process is the reverse process of the encryption which consists of:

- Inverse shift row
- Inverse substitute byte
- Add round key
- Inverse mix columns

There is a variety of round present of key and block within the algorithm. The quantity of rounds depends on the length of key use for Encryption and Decryption.

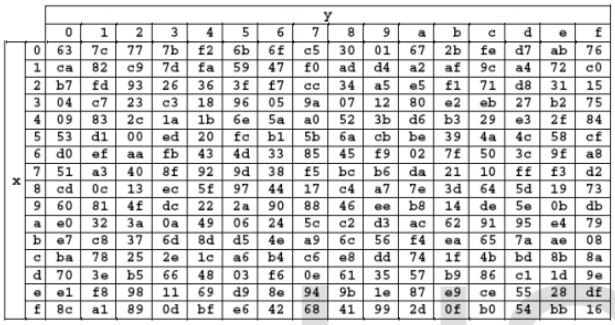
	Key length(word/byte/bits)	Block length(word/byte/bits)	Number of rounds
AE-128	4/16/128	4/16/128	10
AES-192	6/24/192	4/16/128	12
AES-256	8/32/256	4/16/128	14

AES algorithm uses a round function for each its Cipher and Inverse Cipher. This function is composed of 4 different byte-oriented transformations.

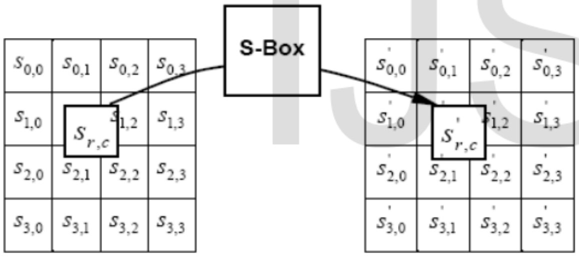
2.1 ENCRYPTION METHOD

2.1.1 SUBSTITUTE BY TRANSFORMATION

The Substitute bytes transformation could be a non-linear byte substitution that operates severally on every byte of the State employing a substitution table S-box. The operation of substitute byte is shown in figure 1.

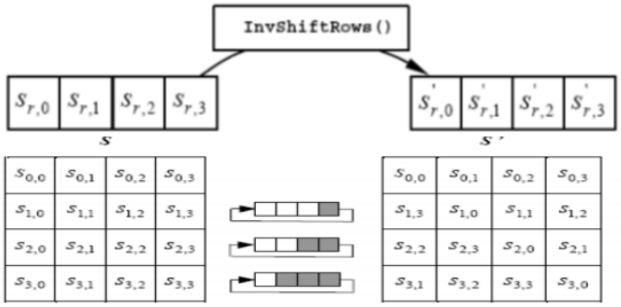


		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3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


S-Box
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{r,c}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s'_{0,0}$ $s'_{0,1}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{r,c}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,1}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,1}$ $s'_{3,2}$ $s'_{3,3}$

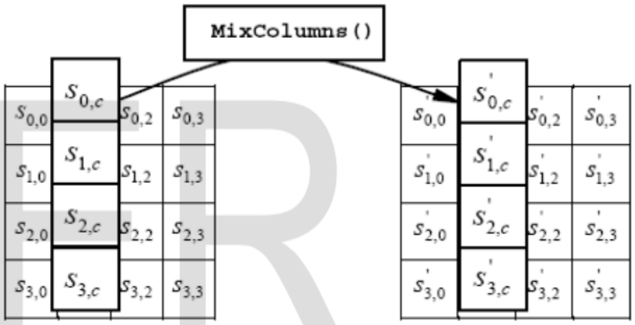
2.1.2 SHIFT ROW TRANSFORMATION

In the Shift Rows transformation, the bytes within the last 3 rows of the State are cyclically shifted over totally different numbers of bytes. The primary row, r = 0, isn't shifted. This has the impact of moving bytes to "lower" positions with in the row whereas the "lowest" bytes wrap around into the "top" of the row.


InvShiftRows()
$s_{r,0}$ $s_{r,1}$ $s_{r,2}$ $s_{r,3}$
S
$s'_{r,0}$ $s'_{r,1}$ $s'_{r,2}$ $s'_{r,3}$
S'
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,3}$ $s_{1,0}$ $s_{1,1}$ $s_{1,2}$
$s_{2,2}$ $s_{2,3}$ $s_{2,0}$ $s_{2,1}$
$s_{3,1}$ $s_{3,2}$ $s_{3,3}$ $s_{3,0}$

2.1.3 MIX COLUMNS TRANSFORMATION

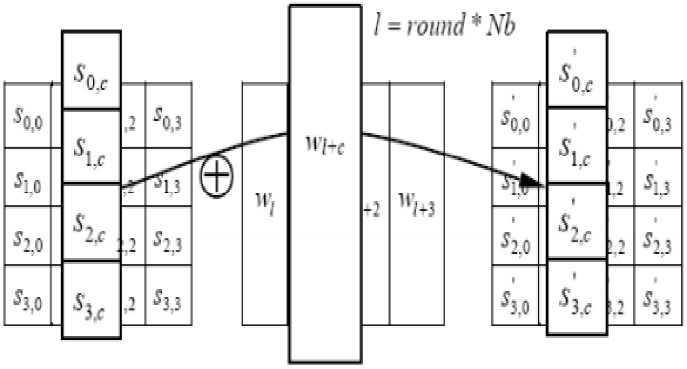
The Mix Columns transformation operates on the State column-by-column, treating every column as a four-term polynomial. The columns are considered as polynomials over GF(2^8) and increased modulo x 4 + 1 with a fixed polynomial a(x), given by a(x) = {03}x ^3 + {01}x^ 2 + {01}x + {02} . The resultant columns are shown within the figure below. This is often operation of mix columns.


MixColumns()
$s_{0,0}$ $s_{0,c}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,c}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,c}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,c}$ $s_{3,2}$ $s_{3,3}$
$s'_{0,0}$ $s'_{0,c}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,c}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,c}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,c}$ $s'_{3,2}$ $s'_{3,3}$

2.1.4 ADD ROUND KEY TRANSFORMATION

In the Add Round Key transformation, a Round Key is superimposed to the State by a straight forward bitwise XOR operation. The Round Key is derived from the Cipher key by suggests that of key schedule method. The State and Round Key are of the constant size and to get the following State an XOR operation is completed per element:

$$b\ (i, j) = a\ (i, j) \oplus k\ (i, j)$$

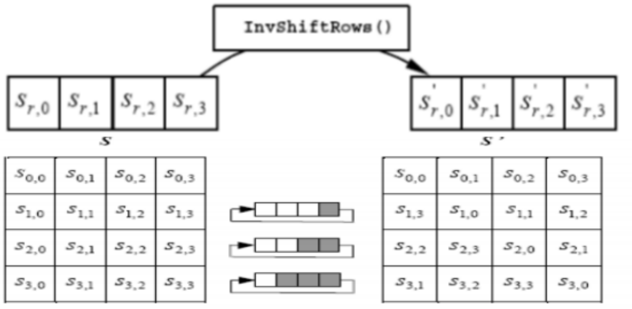

$l = round * Nb$
$s_{0,0}$ $s_{0,c}$ $s_{0,3}$
$s_{1,0}$ $s_{1,c}$ $s_{1,3}$
$s_{2,0}$ $s_{2,c}$ $s_{2,3}$
$s_{3,0}$ $s_{3,c}$ $s_{3,3}$
$\oplus$
w_l
w_{l+c}
w_{l+3}
$s'_{0,0}$ $s'_{0,c}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,c}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,c}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,c}$ $s'_{3,3}$

2.2.1 DECRYPTION METHOD

2.2.1.1 INVERSE SHIFT ROW TRANFORMATION

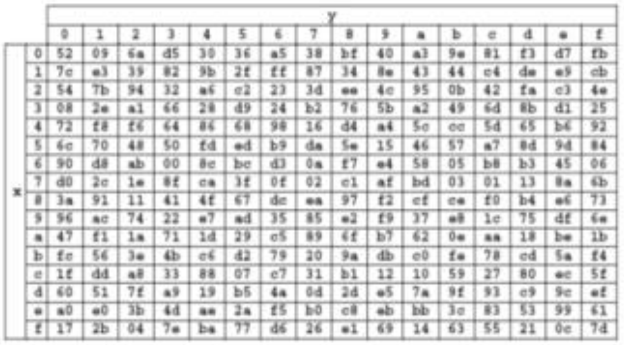
Inverse Shift Rows is the inverse of the Shift Rows transformation. The bytes within the last 3 rows of the State are cyclically shifted over totally different numbers of bytes. The primary row, r = 0, isn't shifted. The bottom 3 rows are cyclically shifted by Nb-shift (r, Nb) bytes, wherever the shift value shift(r,Nb) depends on the row number.

.


InvShiftRows()
$S_{r,0}$ $S_{r,1}$ $S_{r,2}$ $S_{r,3}$
S
$S'_{r,0}$ $S'_{r,1}$ $S'_{r,2}$ $S'_{r,3}$
S'
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,3}$ $s_{1,0}$ $s_{1,1}$ $s_{1,2}$
$s_{2,2}$ $s_{2,3}$ $s_{2,0}$ $s_{2,1}$
$s_{3,1}$ $s_{3,2}$ $s_{3,3}$ $s_{3,0}$

2.2.1.2 INVERSE SUBSTITUTE BYTE TRANSFORMATION

Inverse Substitute Bytes is that the inverse of the byte substitution transformation, within which the inverse S-box is applied to every byte of the State. It’s reverse method of Substitute byte transform. This is often obtained by applying the inverse of the affine transformation followed by taking the opposite in GF (2^8). There is an inverse s-box table for substitute



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

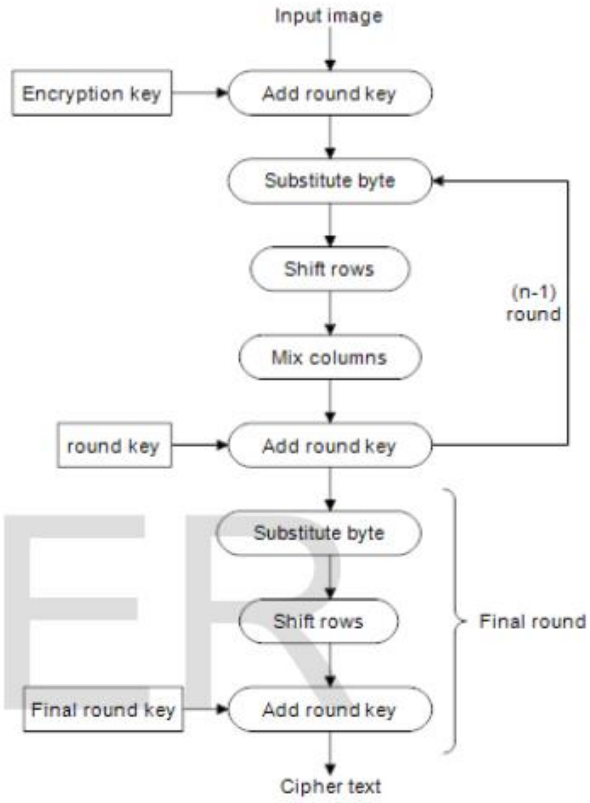
2.2.1.3 INVERSE MIX COLMN TRANFORMATION

Inverse Mix Columns is that the inverse of the Mix Columns transformation. Inverse Mix Columns operates on the State column-by-column, treating every column as a four-term polynomial. The columns are considered as polynomials over GF(2^8) and multiplied modulo x^ 4 + 1 with a hard and fast polynomial (x), given by

(a*-1)(x) = {0b}x^3 + {0d}x^ 2 + {09}x + {0e}

3.IMPLEMENTATION

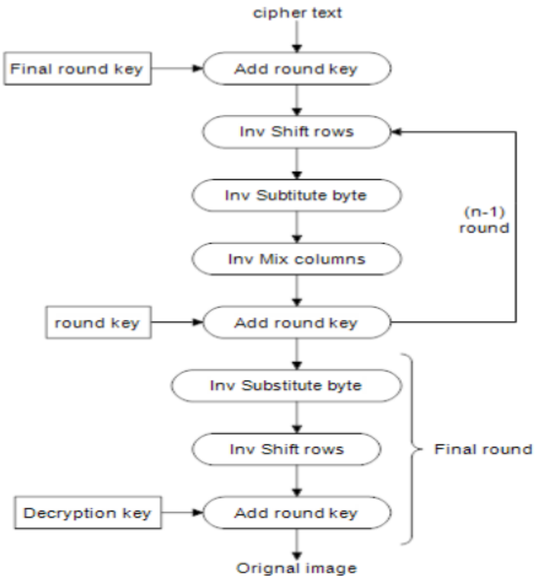
3.1 ENCRYPTION ALGORITHM



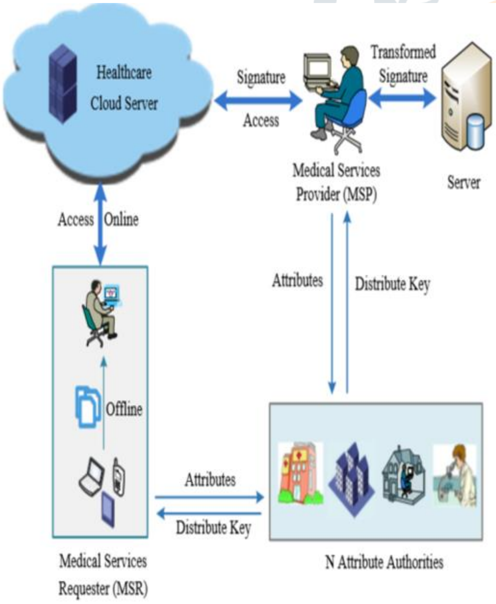
The implementation of the AES-128 encryption and decryption algorithm with the assistance of certain software is completed. During which the input is an image and the key in hexadecimal format and the output is the same as that of input image. For encryption method first, dividing image and creating it 4*4 byte state i.e. matrix format. Calculate the number of rounds based on the key Size and expand the key using our key schedule. And there are (n-1) rounds performed which are substitute byte, shift rows, mix columns and add round key. The final round “n” does not consist of mix column in the iteration. Figure 6 shows the flow of algorithm.

3.2 DECRYPTION ALGORITHM

The AES decryption method is that the lapel method that of the encryption method. The above figure shows flow of the AES decryption algorithm. That encompass of cipher text as the input, the key is constant for decryption method which for encryption. In case of decryption the inverse substitute byte, inverse shift rows and therefor inverse mix columns are to be enforced. Whereas the add round key remains similar for Image Encryption and Decryption. The original pictures may also be fully reconstructed with no distortion. It’s shows that the algorithms have very large security key space and can withstand most typically attacks like the brute force attack, cipher attacks and plaintext attacks



4. SYSTEM ARCHITECTURE



5. DATAFLOW DIAGRAM

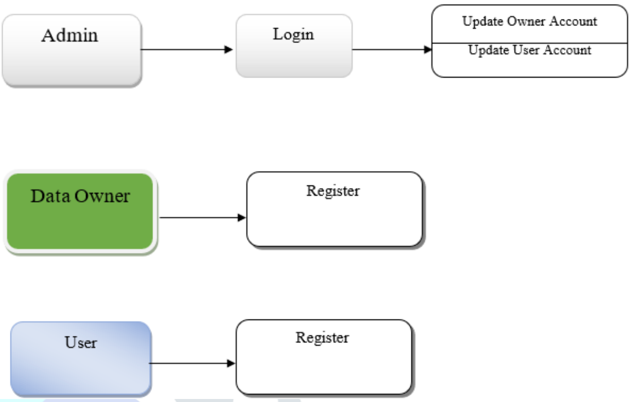
1. The DFD is also referred to as bubble chart. It is an easy graphical formalism that can be used to represent a system in terms of input files to the system, varied process distributed on this data, and therefore the output data is generated by this system.

2.The data flow diagram (DFD) is one in all foremost necessary modelling tools. It is used to model the system components. These elements are the system method, the data employed by the method, associate degree external entity that interacts with the system and therefore the information flows in the system.

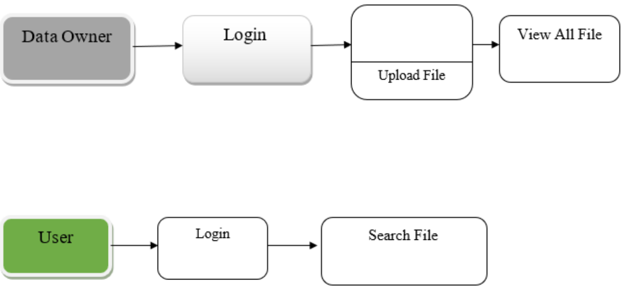
3.DFD shows how the information moves through the system and how the way it is changed by a series of transformations. It is a graphical technique that depicts data flow and therefore the transformations that are applied as data moves from input to output.

4. DFD is additionally called as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD is also partitioned into levels that represent increasing information flow and functional detail.

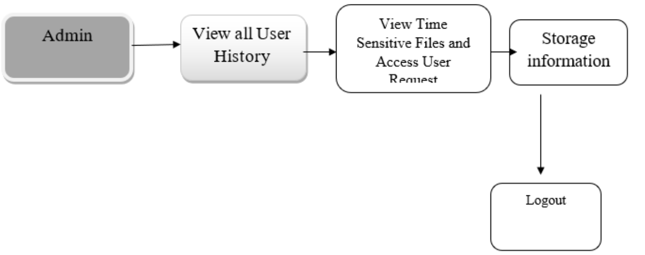
LEVEL 0:

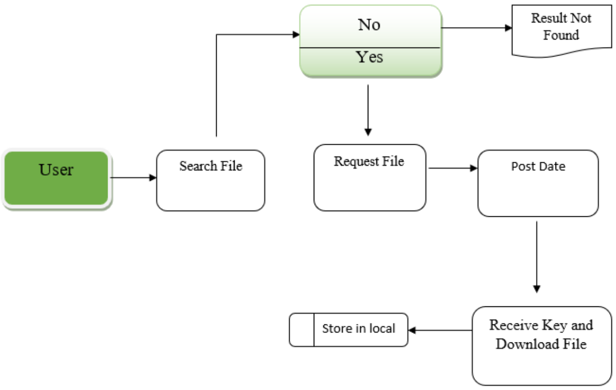


LEVEL 1:



LEVEL 2:





6. CONCLUSION

In this paper, we proposed a lightweight and privacy-preserving medical services access scheme based Advanced Encryption Standard (AES) for healthcare cloud, the security analysis shows that LPP-MSA meets the requirements of unforgeability, anonymity and collusion resistance. Furthermore, the performance analysis of LPP-MSA and several existing schemes shows that LPP-MSA has high computational efficiency. Therefore, LPP-MSA is more suitable for large scale remote medical services access in healthcare cloud system. You can keep track of your health records. Permission is given to only authorized users; they can modify your data.

8.REFERENCES

[1] W. Tang, J. Ren, and Y. Zhang, “Enabling trusted and privacy-preserving healthcare services in social media health networks,” IEEE Transactions on Multimedia, vol. 21, no. 3, pp. 579–590, 2019.

[2] X. Du and H. H. Chen, “Security in wireless sensor networks,” IEEE Wireless Communications, vol. 15, no. 4, pp. 60–66, 2008.

[3] W. Tang, J. Ren, and Y. Zhang, “Enabling trusted and privacy-preserving healthcare services in social media health networks,” IEEE Transactions on Multimedia, vol. 21, no. 3, pp. 579–590, 2019

[4] H. K. Maji, M. Prabhakaran, and M. Rosulek, “Attribute-based signatures,” in Proc. of Cryptographers’ Track at the RSA conference, 2011, pp. 376– 392.

[5] X. Chen, J. Li, X. Huang, J. Li, Y. Xiang, and D. S. Wong, “Secure outsourced attribute-based signatures,” IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 12, pp. 3285–3294, 2014.

[6] S. T. Ali and B. Amberker, “Attribute-based group signature without random oracles with attribute anonymity,” International Journal of Information and Computer Security, vol. 6, no. 2, pp. 109–132, 2014.

[7] X. Du, M. Guizani, Y. Xiao, and H. H. Chen, “A routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks,” IEEE Transactions on Wireless Communications, vol.8, no. 3, pp. 1223–1229, 2009.