CRYPTOGRAPHY SECURITY TO INTERNET OF THING

Mr. Vivekanand Verma, Dept. of Information Technology

Dr. C.V. Raman University, Bilaspur

ABSTRACT-Increase in the use of the network there is a need for the data to be secured, to make the data secure there exist number of algorithms like AES (Advance Encryption Standard) and IDEA (International Data Encryption Algorithm). It secures data using cryptography. Internetof thing (IOT) to the cryptography is new domain which is emerging widely.IOT is defined as the controllingany data from anywhere using internet. Thus, IOT require security. In this paper discuss about how data can be secured for IOT using the methods of cryptography[1].

KEYWORDS-Advance Encryption Standard, Internet of thing, Data encryption, cryptography.

INTRODUCTION-IOT security[2][3] involves securing the data and securing the data and uploading it to cloud. Encrypting the data's using AES-128 algorithm[4].This involves following steps-

S-box, Shift rows, Mix Column, Add round key.

On following the steps for 10 rounds we get the encrypted data. This encrypted data is uploaded to the cloud using IOT development board. Here Intel Galileo board is used. Similarly to retrieve the data inverse AES-128 algorithm is used which involves following steps-

Inverse S-box, Inverse Shift rows,Inverse Mix Column,Inverse Add round key.

On following the steps for 10 rounds we get the decrypted data i.e. original data. Fig.1 gives the block diagram overview.

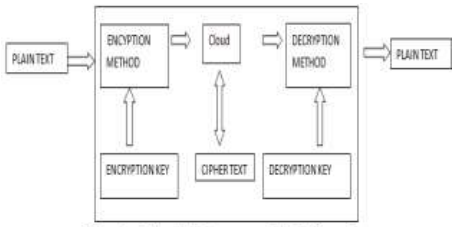


Fig. 1. Block Diagram of IoT Security

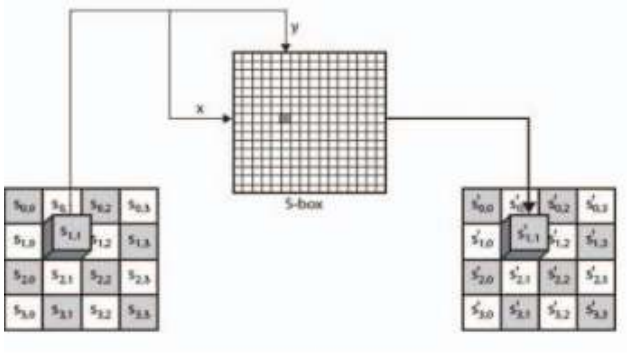
AES-128 ALGORITHM- Advanced Encryption Algorithm is also known as Rilndael was established by U.S National Institute of Standards andTechnology (NIST) in 2001[5]. AES uses 3 types of data which are- 128 bits with 10 rounds, 192 bits with 12 rounds, 256 bits with 14 rounds. AES-128 uses 10 rounds each with 4 steps[6]. Thus, data form a 4x4 matrix with each

elements being 8bits. Data input is a plain text giving an encrypted data while for the decryption input data is a cipher data with original data as output.

S- BOX- In this each element in the plain text is replaced by the element of the s-box, where s-box is a 16x16 matrix with each element with 8 bits. Replacement of the data is done in such a way that first8 bits represent row and next 8 bit represent column.

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

Fig. 2. S-Box



- SHIFT ROWS- This step is applied to the row of the matrix, in thisfirst row remain unchanged while second row get left shift by 1 cyclically, third row get left shift by 2 cyclically, and so on. Which give a 4x4 shifted matrix with each elements of 8 bits.

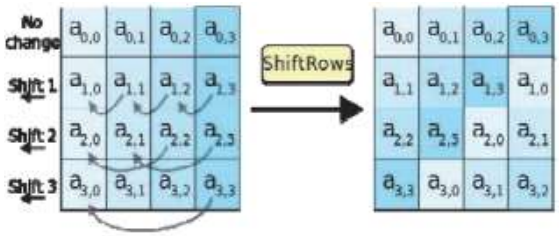


Fig. 4. Shift Row Overview

- MIX COLUMN- This step involves a matrix multiplication with a polynomial. Which give a 4x4 matrix of with each element of 8 bits. For 0x01 multiplication there is no change to the value while multiplication with 0x02 shift data to left by 1 bit, if original data has high bit at MSB (Most significant bit) then sifted data need to be XORed with 0x1b and so on

$$c(x) = \begin{bmatrix} 0x02 & 0x03 & 0x01 & 0x01 \\ 0x01 & 0x02 & 0x03 & 0x01 \\ 0x01 & 0x01 & 0x02 & 0x03 \\ 0x03 & 0x01 & 0x01 & 0x02 \end{bmatrix}$$

Fig. 5. c(x) matrix

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

Fig. 6. Mix Column Step

- ADD ROUND KEYS- This generate key having XOR operations of the mix column matrix with generated for corresponding round gives the output for the current rounds.

The output of this round steps after 10 rounds provides the encrypted data. The key for each round is derived from the original key using Rijndael's key schedule.

INVERSE AES-128 ALGORITHM- Input to the inverse algorithm is the encrypted data the output is plain text or original data. Inverse AES-128 algorithm involves the following steps-

- INVERSE S-BOX- it works same as s-box, steps provide and output of 4x4 matrix with each element of 8 bits.

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

Fig. 7. Inverse S-box

- INVERSE SHIFT ROW- In this first row is kept same, while second rows is shifted towards 1 by right cyclically. The third row is shifted towards right by 2 cyclically and so on.
- INVERSE MIX COLUMN- it is similar to the Mix column step as shown in figure.
- INVERSE ADD ROUND KEY- In this a XOR of keys for the corresponding round with the inverse mix column matrix gives an output. Again key generation involves Rilndael key schedule.

$$c(x) = \begin{bmatrix} 0x0e & 0x0b & 0x0d & 0x09 \\ 0x09 & 0x0e & 0x0b & 0x0d \\ 0x0d & 0x09 & 0x0e & 0x0b \\ 0x0b & 0x0d & 0x09 & 0x0e \end{bmatrix}$$

Fig. 8. Inverse Mix Column Step

APPLICATION- Bird Hit Probability-In this using a camera we can capture the image of bird. Using image processing we can detect the position of bird using x, y and z coordinates. Thus this plain data is encrypted and uploaded to cloud. At other end, the received data is decrypted and use those data to avoid the collision of airplane and birds.

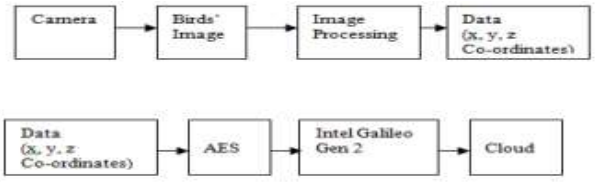


Fig. 9. Uploading encrypted data to Cloud

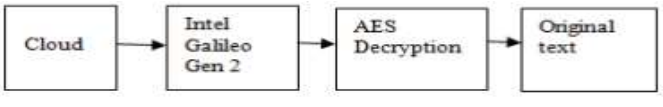


Fig. 10. Data retrieval from Cloud using inverse AES

RESULTS AND APPLICATION-

Fig. 11. Image when bird is detected

Fig. 12. Data in terms of x, y and z co-ordinates

CONCLUSION AND FUTURE SCOPE- The data encryption and decryption is done for 128 bits. To make the data more secure. In bird hit probability, since the movement of the bird is not stationary, capturing the image, process it, encrypt it and upload it to cloud would consume time. To overcome thisblack box can be made which receive data as a plain text and give the encrypted output which can be used for many TOI applications.

REFERENCES-

[1] Y. Peng, W. Zhao, F. Xie, Z. H. Dai, Y. Gao, and D. Q. Chen, “Secure cloud storage based on cryptographic techniques,” *J. China Univ. Posts Telecommun.*, 2012.

[2] M. A. M. Sadeeq, S. R. M. Zeebaree, R. Qashi, S. H. Ahmed, and K. Jacksi, “Internet of Things Security: A Survey,” in *ICOASE 2018 - International Conference on Advanced Science and Engineering*, 2018.

[3] A. Ukil, J. Sen, and S. Koilakonda, “Embedded security for internet of things,” in *Proceedings - 2011 2nd National Conference on Emerging Trends and Applications in Computer Science, NCETACS-2011*, 2011.

[4] S. Cirani, G. Ferrari, and L. Veltri, “Enforcing security mechanisms in the IP-based internet of things: An algorithmic overview,” *Algorithms*, 2013.

[5] D. Mendez Mena, I. Papapanagiotou, and B. Yang, “Internet of things: Survey on security,” *Information Security Journal*. 2018.

[6] D. Ganguly and S. Lahiri, “Cryptography and Network Security,” in *Network and Application Security*, 2011.