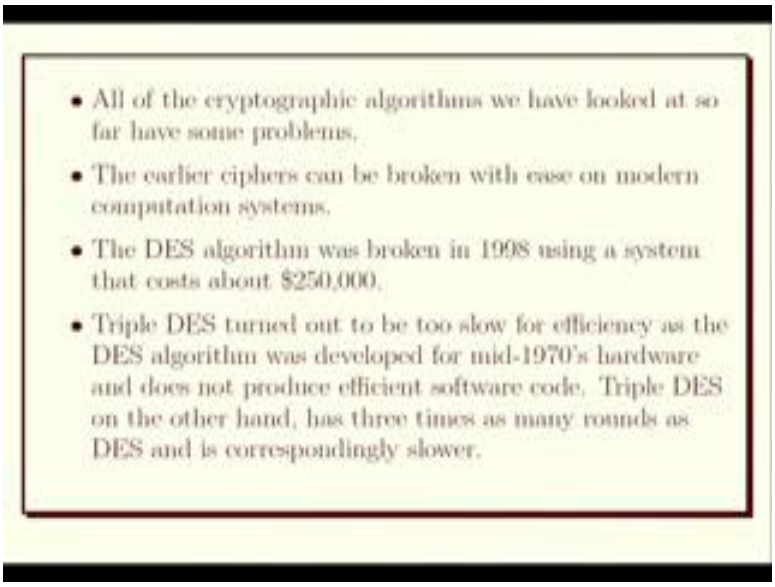
Internetwork Security
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture - 21
Advanced Encryption Standard (AES)

So, we will talk about another block cipher which is called, which is the Advance Encryption Standard or AES.

(Refer Slide Time: 00:31)

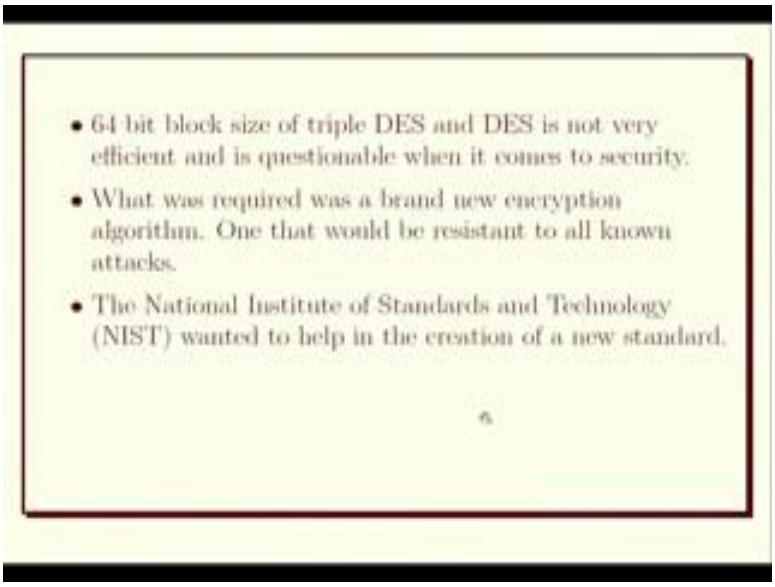


So, can you please go to the slide? So, so far all the cryptographic algorithm we have looked as so far has some problem like we have seen the classical cryptosystem like shift cipher, Caesar cipher those are earlier cipher. So, those are broking under the, if we have modern computer speed, so those are not secured. Even we have looked at we have looked at the DES which is also broken even by the generic attack exhaustive search or time of attack this is by far electronics founder foundation they made a hardware which can crack the DES in less than three days and with a cost of this. So, this is an exhaustive search attack and also DES can, on DES we can one can mound the non generic attack like differential cryptanalysis linear cryptanalysis some variant of differential attack, those attacks are there for DES.

So, DES is not secured, so we need to have alternate for the standard. So, we tried with the triple DES people tried with the triple DES, but triple DES is basically made from the

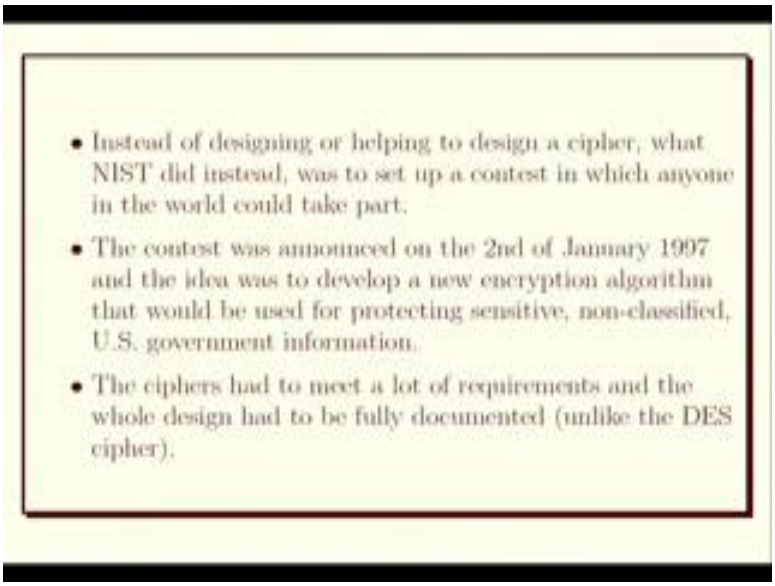
DES itself which is basically three times DES, two times encryption, one time decryption and for decryption triple DES two time decryption one time encryption. So, each DES is 16 rounds. So, total round in the triple DES is 48 rounds which is huge, which will make our algorithm very slow.

(Refer Slide Time: 02:17)



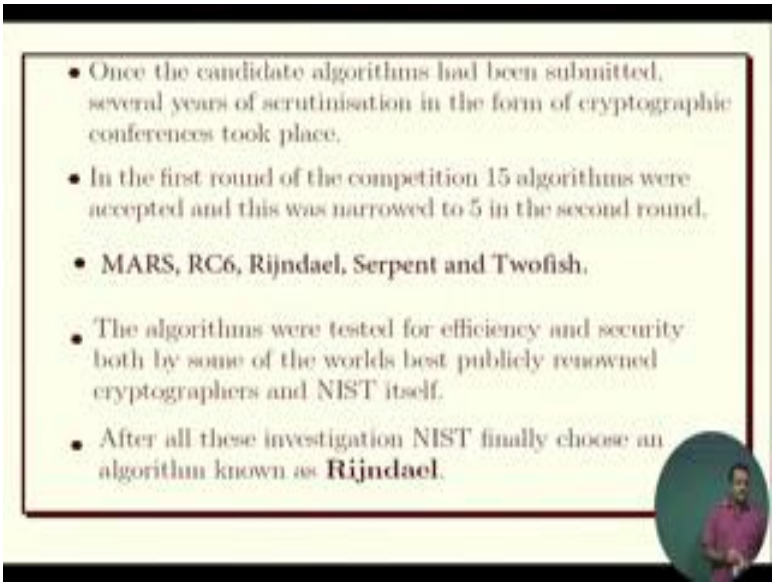
So, then we and also DES is we have seen it is a problem with the block size it is a 64 bit block and so, now we need a brand new encryption algorithm, we need a block cipher, new block cipher which is supposed to be secured under the all existing attack. So, what NIST did? So instead of developing a new cipher by themselves, what they did they announced a competition worldwide competition and in that competition anybody can participate, so that was in 1997.

(Refer Slide Time: 02:51)



So, they announced a worldwide competition and they welcomed all the cryptographer worldwide to submit their block cipher design. So, those who are working on the block cipher or encrypt system. So, they just jump into that competition and they designed and, this is the first and they fixed some requirement this cipher, this submitted cipher should be OLE documented unlike the DES because DES we have, we do not know where from this boxes are coming they have just given the table, it is a 4 by 16 table, but there is no really information how these datas are coming. So, they want, the NIST want a full documentation for the new design of block cipher.

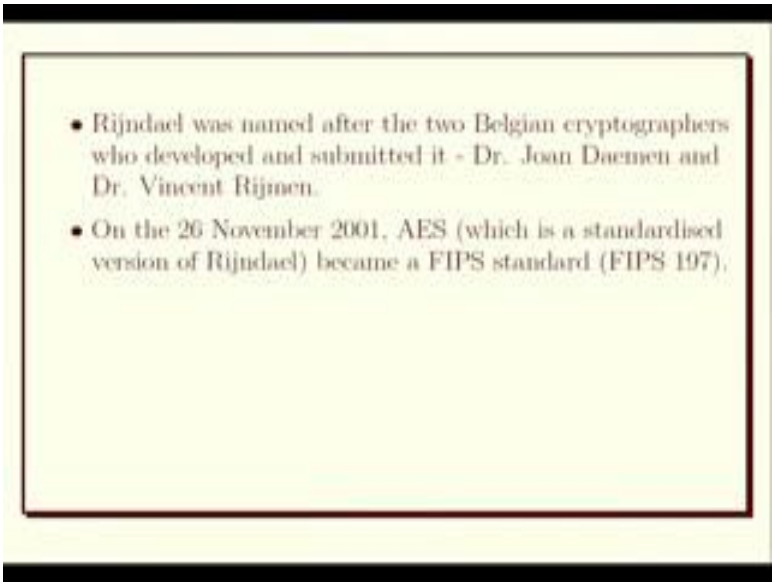
(Refer Slide Time: 04:00)



So, in the first round they got many submissions and from that, after first round they selected 15 algorithm were accepted; that means, 15 algorithm or other algorithm was having some problem, even this was public.

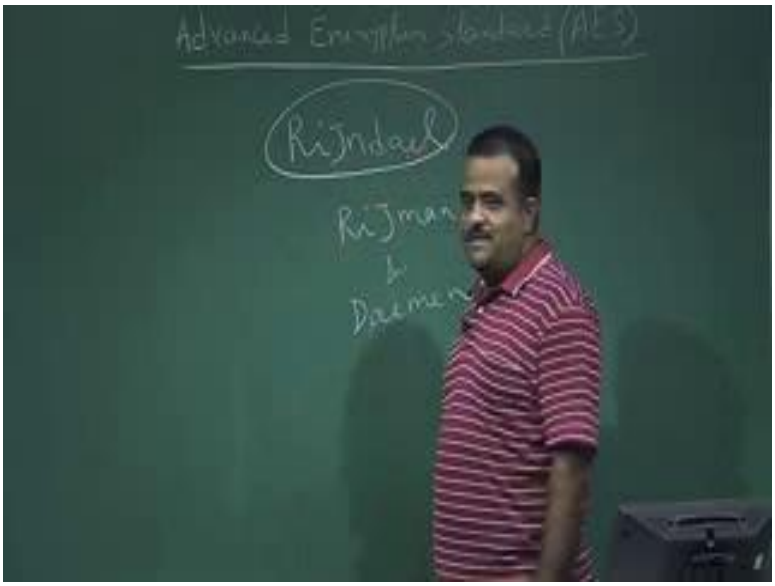
So, anybody can see the algorithm and anybody can comment on this, anybody can attack on this. So, that way it is open competition. So, that is the 15 cipher was selected after first round then after further investigation and after further attack in the second year, at the second round there was 5 cipher which was selected and those names are basically MARS, RC4, Rijndael, Serpent and Twofish. So, these are the cipher which was selected after the second round of this competition. So, again in their final round among these cipher, among these 5 cipher finally, the Rijndael was selected the name of the block cipher is Rijndael and this was selected because other cipher may be people got some attacks on these some faults, may be not OLE documented.

(Refer Slide Time: 05:34)



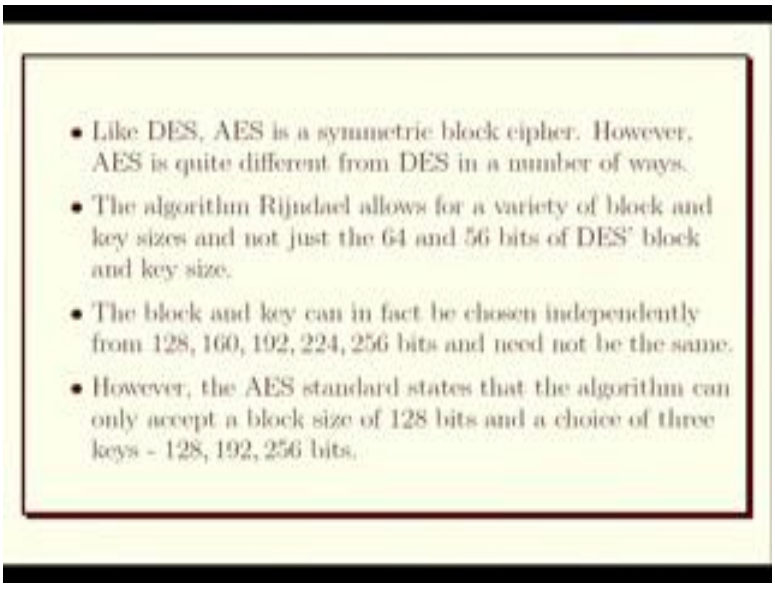
So, finally, this Rijndael competitor, I mean Rijndael is the - The AES, the advance encryption substandard. So, and this Rijndael name came from the name of two cryptographer who designed this cipher this is the two Belgium cryptographer Joan Daemen and the Vincent Rijmen. So, based on their name, it is the name of the cipher is basically Rijndael.

(Refer Slide Time: 05:57)



So, may be this is coming from say Rijman one person and Daemen sorry, d a e m e n. So, based on their name Vincent Rijmen and Joan Daemen, based on their name they proposed this name Rijndael.

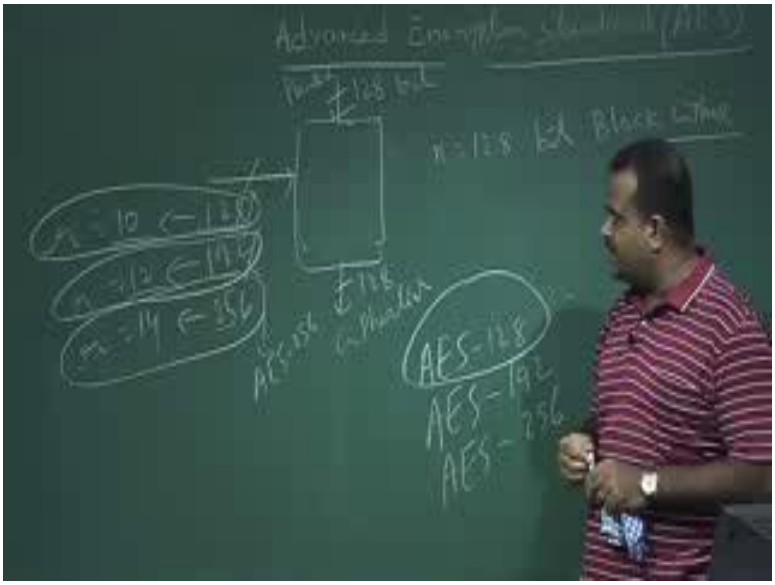
(Refer Slide Time: 06:49)



So, Rijndael is the AES which was make standardized in 2001 by NIST. So, still now this is the AES this is the standard for block cipher. So, basically it is a block cipher, so it is a symmetric key primitive, but it is quite different than the DES the earlier block

cipher DES. So, how it is different from DES? DES was having key size 56 bit affected key size, now here we have 3 variant of the Rijndael.

(Refer Slide Time: 07:23)



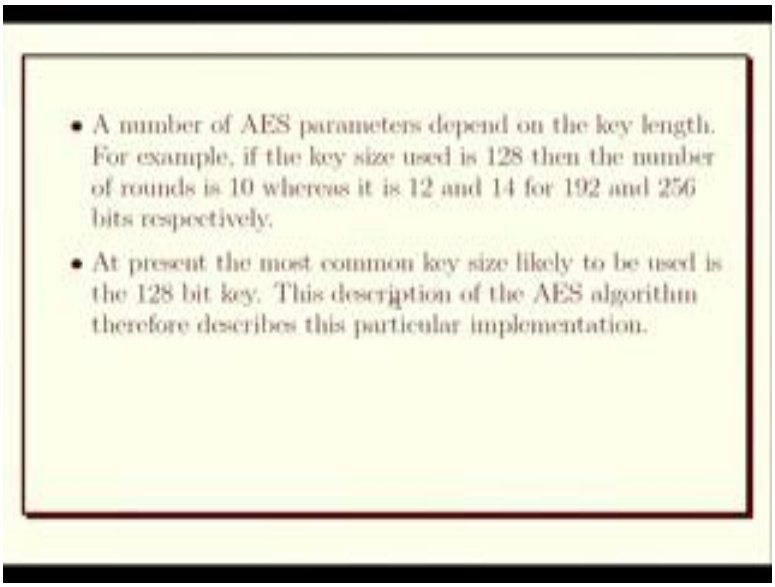
So, Rijndael as 3 versions, so this is basically a block cipher this is Rijndael or the AES. So, block size is 128 bit, so this is a 128 bit, n is 128 bit block cipher, block cipher; so that means, plaintext and ciphertext size is 128 bit, this is ciphertext and this one is plaintext. Now the key size there are 3 variant of this - one is 128 bit key another variant is 192 another variant is 256, now depending on the key size we have the number of rounds for 128 bit key we have r is the number of round is r is equal to 10 rounds, this is one version. Another version is if it is 192 then r is 12 round and if it is 256 then r is 14 round.

So, depending on the size of the key we have the different different round. So, these are called, depending on the size of the key these are called AES-128, AES-192, AES-256, but in each of this cases the n is fixed it is a plaintext and ciphertext size is basically 128 that is fixed. But only thing we have three variant depending on the size of the key and the number of round. If the size of the key is 128 then the round is this, this version is called AES-128 and if the size of the key is 192 then the number of round is 12 and this version of AES is called AES-192 and this version of AES is called AES-256.

So, among these this is the most popular one because this has less number of rounds, so it will be more efficient in the terms of competition. So, it will be little faster than others

because others has more round and the key size also less, so if we can achieve the same level of security by using less key size less number of round then obviously, that will be most popular.

(Refer Slide Time: 10:29)

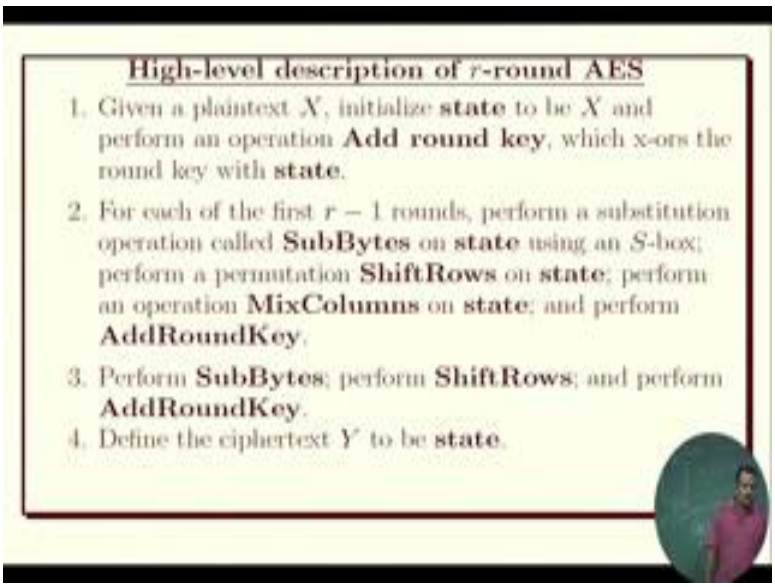


So, can you go to the slide please? So, the number of; so this is the thing, so it depends on this number of AES parameters depend on the key length. For example, the key size used is 128 bit then the number of round is 10 which we have already discussed and if the number of round is 12 then the key size must be 192 and if the number of round is 14 then this. So, among these most common key size is we take 128 bit.

(Refer Slide Time: 11:01)

So, Rijndael basically was designed to have the following characteristics. So, it must resist from the all known attack - like generic attack, like linear attack, differential attack, all version of the differential attack, like boomerang attack, so all these attacks. So, it should resist from the all known attack and it should be fast. So, computationally it should be fast, speed and speed should be fast and it should widely use and the design should be simple, so that that will make the cipher fast. So, following this characteristics it became the AES.

(Refer Slide Time: 11:54)



This is a high level description of a typical r-round AES. So, basically what we have? We have a plaintext X and plaintext is 128 bit. So, we will just store this plaintext in a state we will come to that how we can make the state, and then we have few operations we have basically 4 operations - one is add round key, it is basically; sorry, we take the input as the plaintext and it is basically taking the round key and it is xoring with the round key, bit wise xoring that is the Add Round Key very simple operation. And the other three operations are SubByte operations this is the S-box operation. So, we take a byte, this state each state is a byte. So, we take a byte and then we apply the S-box and then we get the output. So, that S-box is taking 8 bit input and giving as 8 bit output.

(Refer Slide Time: 13:12)

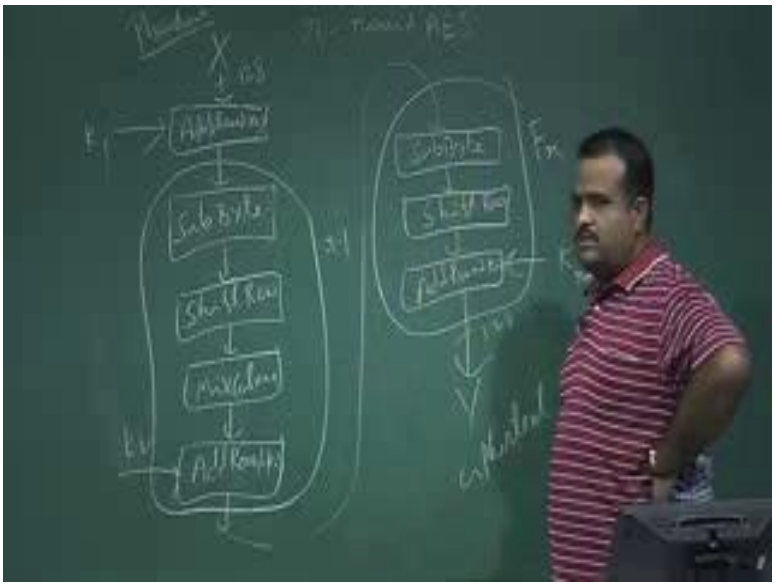
So, the operation basically first one is mixed Add round key. So, this is basically adding the round key. So, bit wise xoring the round key with the input, first input is the plaintext, but later on intermediate value. Then the SubByte operation this is basically S-box, so we have 8 by 8 S-box, we have 8 bit input 8 bit output and this is taking 8 bit means 8 byte, this is byte wise operation. So, we take 8 bit and it will replace by 8 bit. So, this is the SubByte operation and we will come to that AES SubByte operation and then the ShiftRow.

ShiftRow, we are just basically shifting the row, we store the input in a matrix in a 4 by 4 state which is called state, I will come to that how what is the state; ShiftRow and then MixColumn. So, these are the basic, we will also discuss what is the MixColumn. So,

these are the basic operation we will use for designing the AES. So, these are the basic operation operations used in AES and we will study each of this operation in details. So, can you go to the slide please?

So, this is the typical r-round AES. So, first we take the plaintext and we store it into a 4 by 4 matrix and each entry is a byte and then we perform the Add round key operation that is the first operation we will do, and then we will perform this SubByte operation ShiftRow operation, MixColumn operation, again AddRoundKey operation. So, these will continue for r minus 1 round and then after that for the last round, it is rth round. So, r many functions are there. So, in the last round what we will do we just apply the SubByte operation, ShiftRow operation, AddRoundKey operation. So, in the last round the MixColumn operation is missing.

(Refer Slide Time: 16:13)

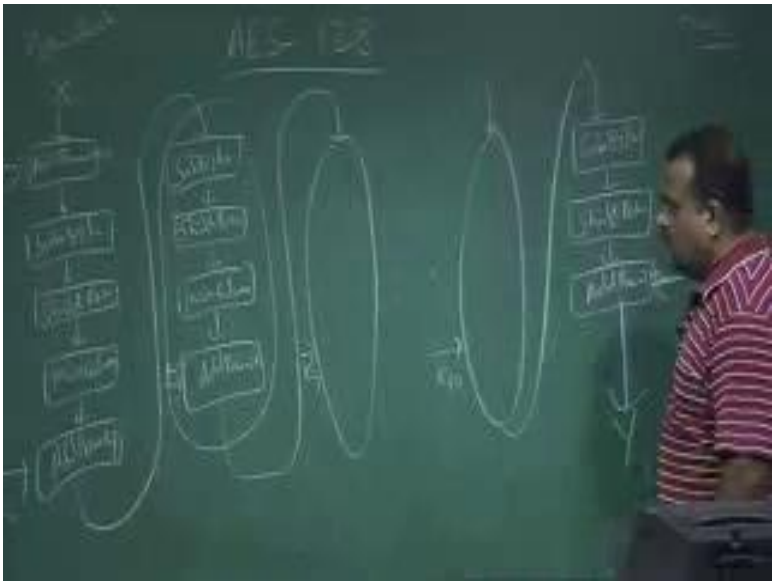


So, basically this is the structure of AES. So, we have the plaintext X. So, this is just first we add a round key AddRoundKey, AddRoundKey. So, for this we need a round key K 1 and then we then we have r minus 1 round of this, so ShiftRow then sorry; this is SubByte, first one is SubByte then the ShiftRow, then MixColumn and then again AddRoundKey. So, for this AddRoundKey we need a round key for next time it is K 2. So, these will repeat. So, this is sort of one round function of AES, this will repeat r minus 1 times. So, again this will perform SubByte, ShiftRow, MixColumn, like this.

Then after that for the last round what we do we just apply SubByte, then ShiftRow and then MixColumn sorry; MixColumn is not there in the last round, that is the difference; so AddRoundKey. So, for last round we need to have if it is r-round AES then we need to have K r plus 1 because we have used one round key in the first slot. So, this is the ciphertext Y and this is the plaintext, both are 128 bit. So, this is a typical r-round AES. So, we have a 128 bit the intex and finally, we are getting a 128 bit; this is the last round, this is the last round F r and the first round is basically this is the extra operation we are doing in the first round, but remaining after that we are just doing this is one round operation, we are doing this for r minus 1 round and then finally, we are going to the last round.

So, now we want to discuss all these details like what are these operations AddRoundKey, ShiftRow, MixColumn, like this. So, let us just quickly talk about we have this is the general structure.

(Refer Slide Time: 19:41)



But we are looking at AES-128 bit or not AES-128, AES 128. So, this version AES we are looking for. So, this is basically a 128 bit block cipher; so that means, plaintext and ciphertext size is 128 bit and it has key size it also 128 bit and it as round is r is 10 round. So, it is basically a 10 round block cipher.

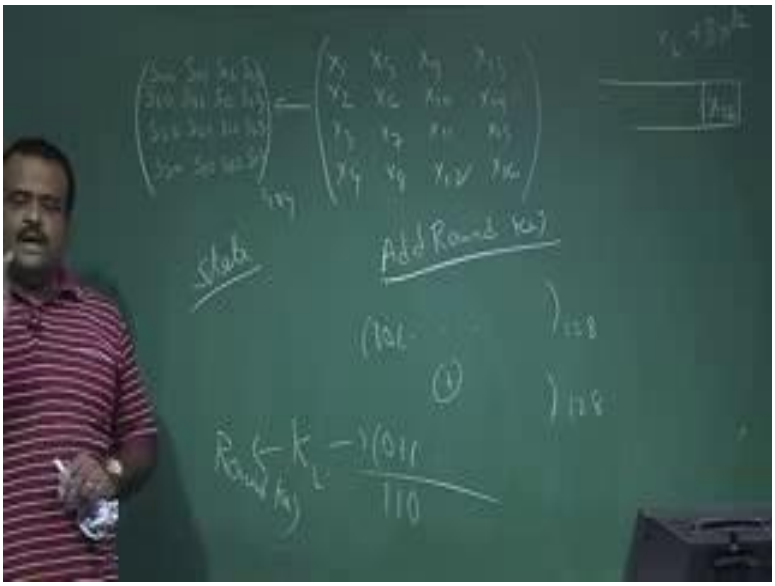
So, what we are doing? This is our X and we are first applying the AddRoundKey with the K 1. So, then we are having this SubByte then mixed SubByte, then ShiftRow,

MixColumn and finally, we have again AddRoundKey. So, this is typically one round AES, these 4 operations. So, this we need a K 2. So, this is sort of first round F 1 then again these we will pass it to SubByte operation then again ShiftRow, so then again MixColumn then again a AddRoundKey. So, add round key. So, here we need a K 2 like this. So, this will continue, this will continue how many times? So, this will continue 9 times.

Again we will do the same thing over here. So, like this, so for this we need to have a K 3 like this. So, we will continue this 9 times and for 9th we have a K 10. So, same thing we will continue. So, in the last round what we do? We have the MixColumn operation is missing we are not doing the MixColumn in the last round. So, in the last round we are just doing SubByte then ShiftRow and then just AddRoundKey; AddRoundKey and here we need to have a round key which is K 11th. So, this is the ciphertext Y. So, this is the typical AES 128 bit block cipher. So, this is the plaintext which is 128 bit and this is the ciphertext. So, now, we will discuss what are these operations basically and how we can have these key scheduling algorithm to get the round keys.

So basically, now, we will talk about AES is a byte wise operation. So, it is taking input in a state, it is state wise operation.

(Refer Slide Time: 23:46)



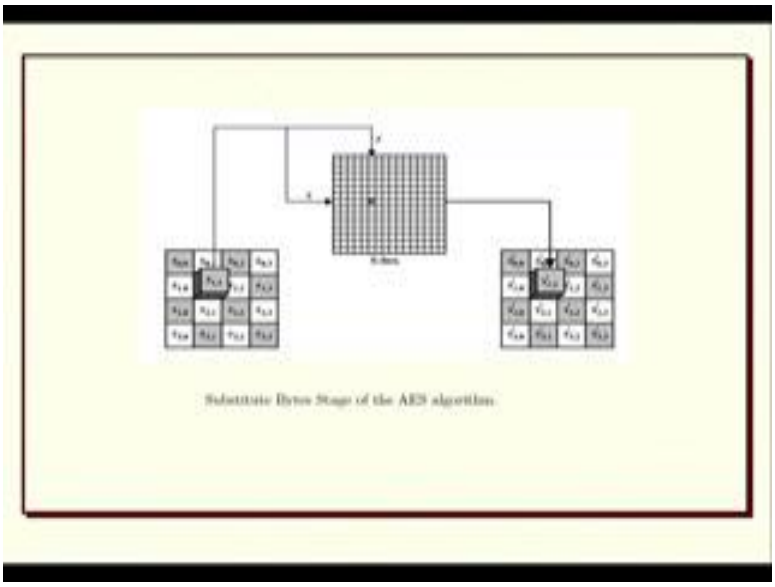
So, state it is a 4 by 4 matrix, we can just denote by S 00, S 01, S 02, S 03, so this is the first row. S 10, S 11, S 12, S 13, S 20, S 21, S 22, S 23, S 30, S 31, S 32, S 33. So, this is

basically a 4 by 4 matrix and each of this is 8 bit. So, it is a byte. So, each of this entry is 8 bit. So, how we can take the input plaintext in a state? So, our plaintext is basically 128 bit this is X, this is 128 bit.

So, what we do? We divide into bytes. So, if we divide into bytes basically we have 16 bytes. So, X 1, X 2 sorry; X 1, X 2, X 3, X 16. So, each X i is a byte or 8 bit. So, now, what we do? We store this here, so we take this like this. So, X 1, X 2, X 3, X 4. So, basically the first entry is X 1 then column wise we are storing then X 5, X 6, X 7, X 8 then X 9, X 10, X 11, X 12 then X 13, X 14, X 15, X 16. So, this way we will form the state, the matrix. So, you are just breaking the plaintext into bytes and each byte is element of this, entry of this matrix. So, this is a 4 by 4 matrix this is called state and AES is; this AES operation the basic operation are state wise operation.

So, now, for the Add Round Key. So, Add Round Key what we are doing? We have a state basically this is again a 128 bit input and suppose our key, round key is also 128 bit this is the round key, so Kth round key, Rth round key. So, Add Round Key is basically bit wise xoring. So, this is a 101 like this and this round keys is 011 like this. So, we are just doing bit wise xoring these x-or with this one, these x-or with this is 1, these x-or with this is 0. So, no carry just bit wise xoring. So, this is basically Add Round Key operation. Very simple operation we are just xoring the, we are adding the round key xoring the round key with the state.

(Refer Slide Time: 27:21)

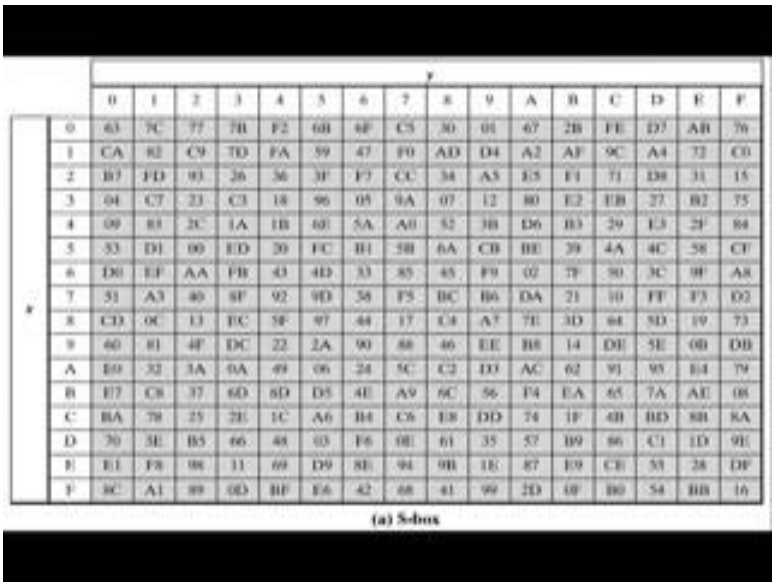


So, now we will talk about this SubByte operation. So, in a SubByte operation, can you go to the slide please? So, this is the state in the SubByte operation what it is taking? It is taking each byte, each entry is a byte. So, it is taking input 8 bit and it is there is a table which is the S-box table for AES and it is doing the table look off and it is giving us the output. So, we have S-box which is 8 bit input and can give us 8 bit output.

(Refer Slide Time: 28:02)

So, this is the SubByte operation for AES. So, basically we have S-box. So, this is the AES S-box. So, this 8 bit, we can take this is x, this is y and x y are hexadecimal bit. So, this is given in the; this can be given in the table, can you go to a slide please.

(Refer Slide Time: 28:34)

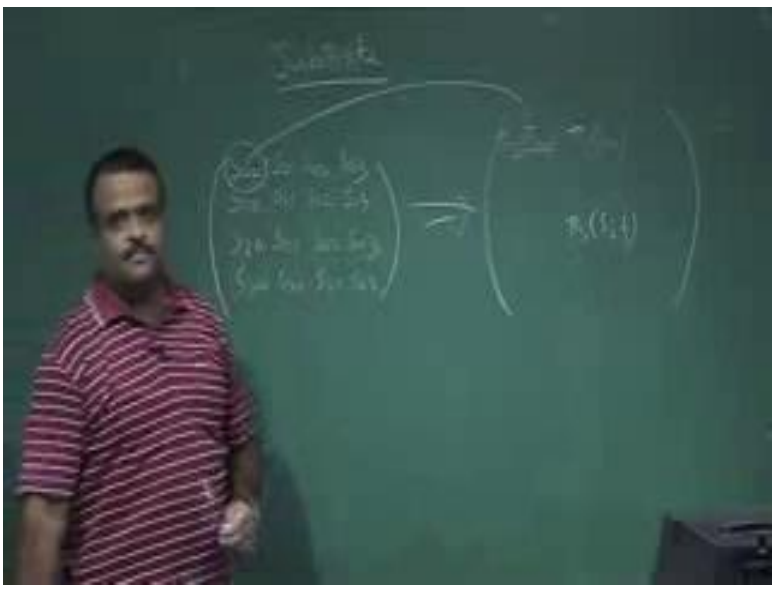


	y															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	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

(a) S-box

So, this is the table for AES S-box. So, this is the x, x is an hexadecimal form and this is the y. So, depending on the value x and y we will go to that particular entry and we will get the output of the S-box. So, suppose x is say, x value is 8 and y value is 7, so we will go to this, so output is C4, C is also in hexadecimal form. So, we have to convert into 4 bit and 4 is also in x form, we have to convert into 4 bit. So, these two will give us 8 bit number. So, we have 8 bit input and 8 bit output. So, this is the way we will just look at the table, so we take a state, we take entry of the state that is the byte. So, these we will do the table look up and we will convert that into pie S ways.

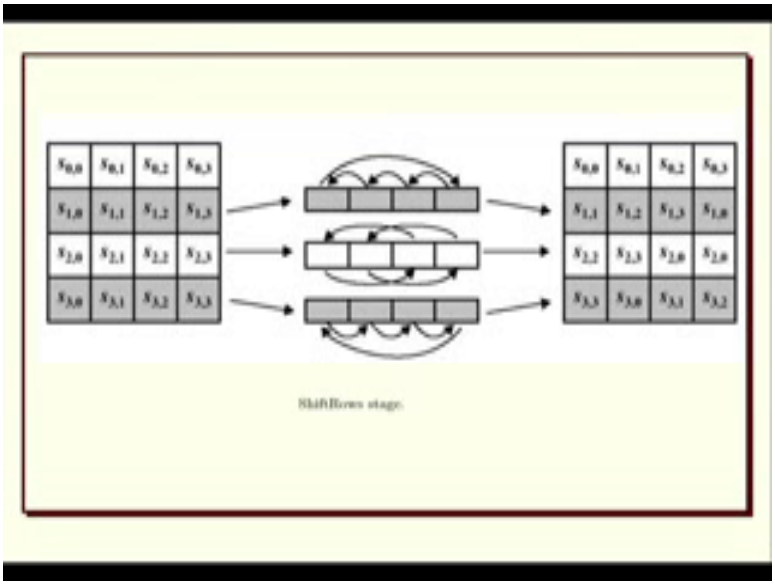
(Refer Slide Time: 29:34)



So, S 00, S 01, S 02, S 03, sorry; S 12, S 13, S 20, S 21, S 22, S 23, S 30, S 31, S 32, S 33. So, this is another SubByte operation we take each of this and we replace by its S-box. So, pie S of S 00, pie S of S 01 are like this. So, in general for i jth element this is pie S of S ij, so like this. So, we take a byte and we apply the S-box we got the again 8 bit. So, that is the SubByte operation.

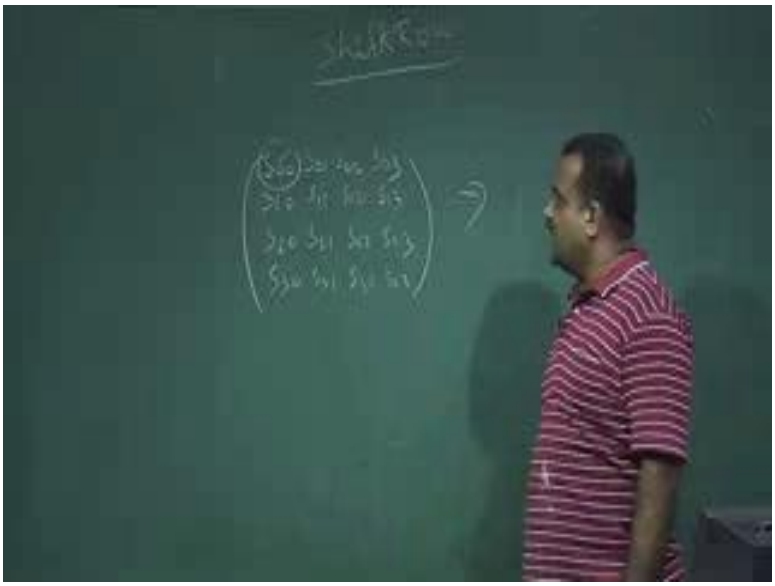
But this AES SubByte is not just a table. So, there is some algebraic structure for this AES SubByte operation we will talk about that in the next class these algebraic structure how we can get the, if we get a, if we give a 8 bit input, how to get the 8 bit output. So, that is basically coming from; we will we have to do some field operation there. So, we will talk about that.

(Refer Slide Time: 31:03)



Now let us talk about the ShiftRow operation. So, in the ShiftRow AES ShiftRow, so this is the ShiftRow operation. So, what we are doing? So, ShiftRow operation for AES.

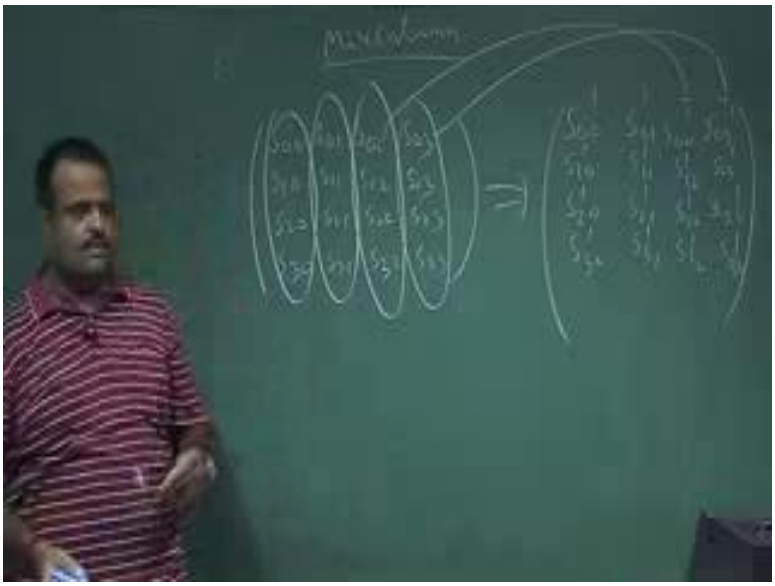
(Refer Slide Time: 31:12)



So, this is a state now we are shifting the rows. So, this is first row, first row will remain unchange and second row we are shifting one times circular shift. So, this is going there, this is going there, this is going here, this is coming here. Third row we are shifting twice this third row, 4th row we are shifting thrice. So, this is the way we are doing. So, this first row remain unchanged, second row we are shifting one position and this next row we are shifting two position circular shift and these row we are shifting three positions. So, that is the way we are shifting. So, this is the ShiftRow operation, basically we are shifting the position of the byte.

So, now, we will talk about MixColumn operation. So, MixColumn operation we will have the; it is taking the column wise. So, S 00, S 01, S 02, S 03, S 10, S 11, S 12, S 13, S 20, S 21, S 30, S 32, S 33. So, in the MixColumn operation it is a column wise operation.

(Refer Slide Time: 32:21)



So, it is taking each of this column and it is replacing by a new column. So, this may be S 00 plus S 00. So, S 20, S 30, again it is taking a column and it is converting into a new column like this; S 21.

So, like this again it is taking a new column this column and it is converting into a new column S 20. So, this is the 12th column 32 and then finally, it is taking the last column it is replacing by new column. So, it is a column wise operation, but these operation involves some field operation we will talk about these in the; S sorry S 33. So, how this column is coming, the new column we will talk about in the next class. So, that is basically field operation we are doing and then in the next class we will also discuss this key scheduling algorithm how we can get the round keys. So, we need 15, we need 11 round keys K 1, K 2, K 11 and each round keys is 128 bit. So, how we can get these round keys from the given secret key which is 128 bit those we will discuss in the next class.

Thank you.