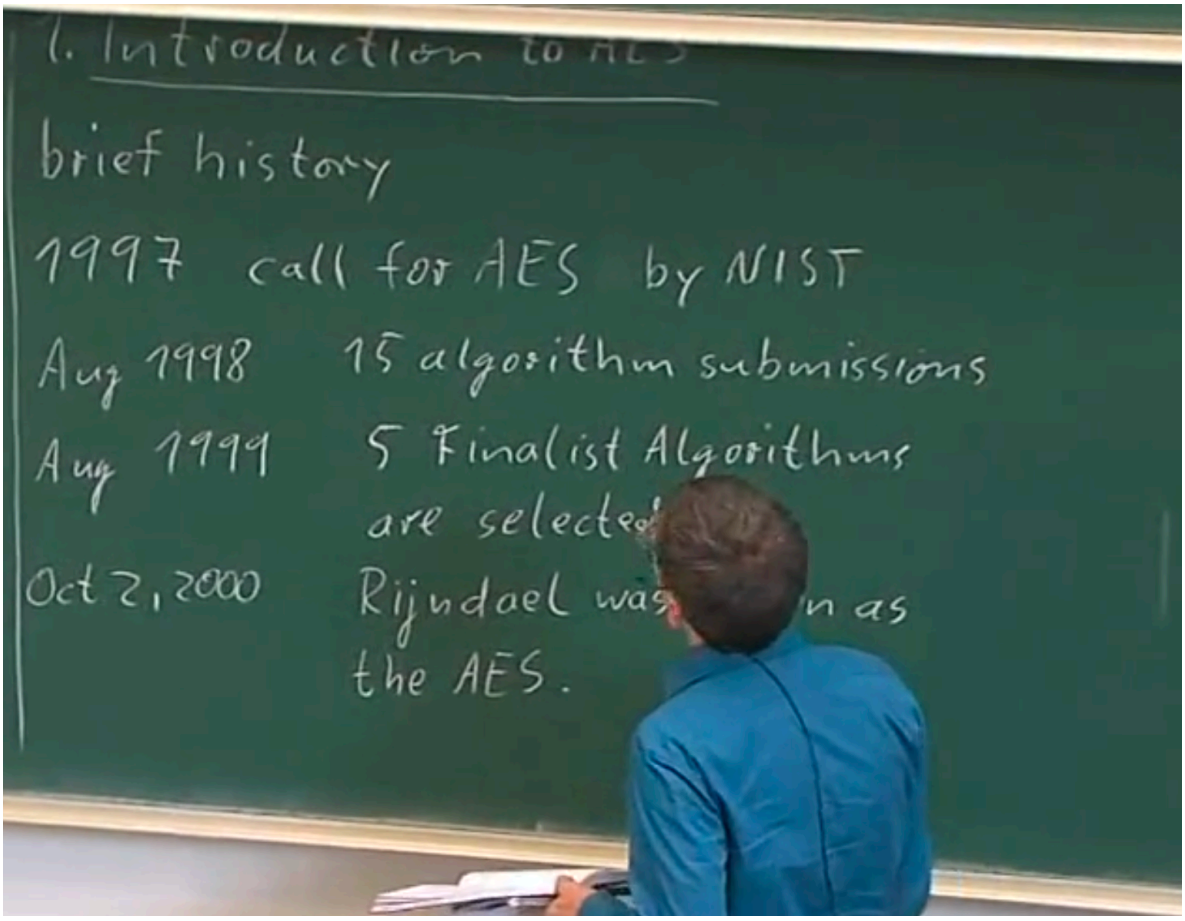
Lecture 8 AES:

Introduction :

History :-

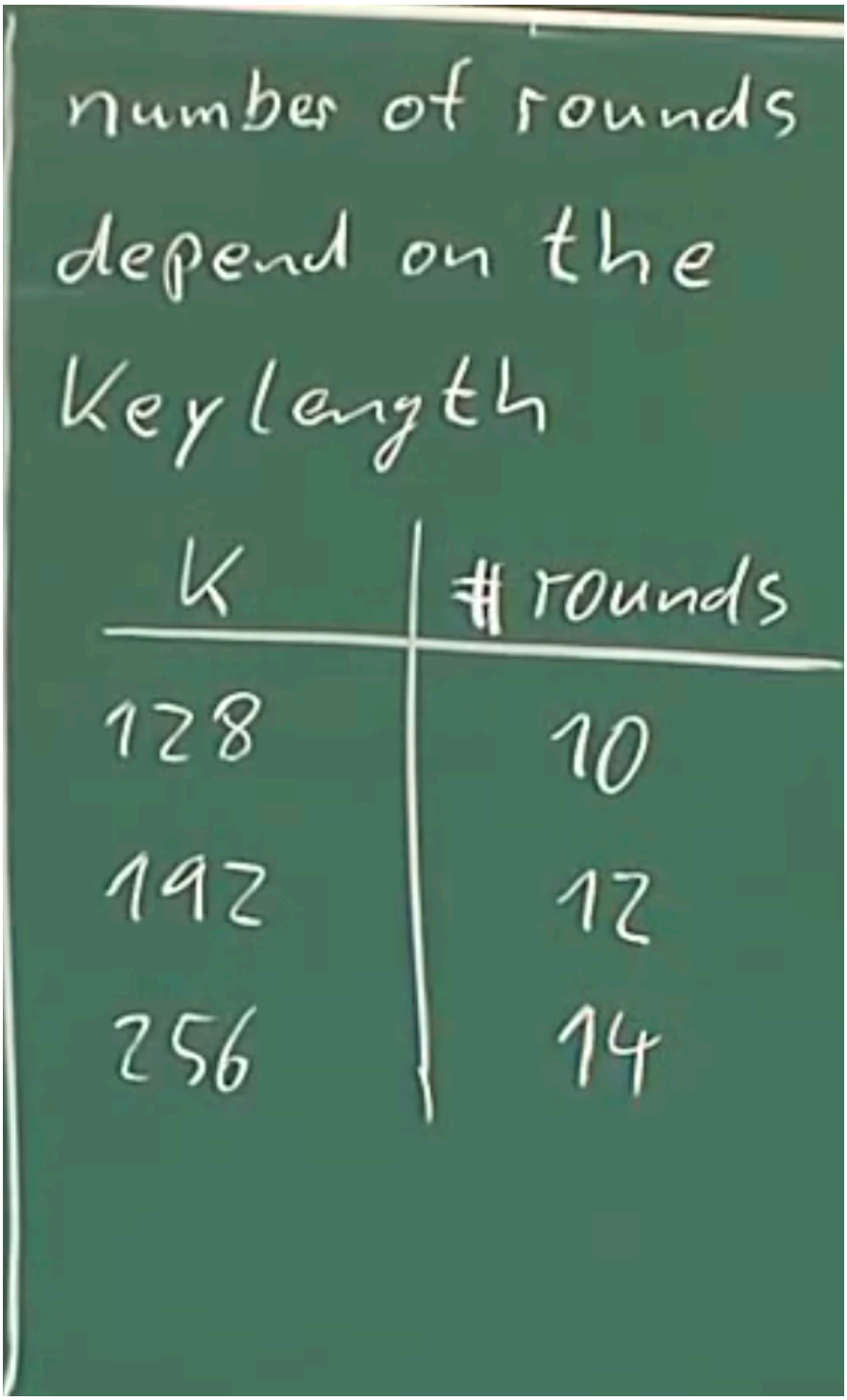


Other 4 finalist's were Bruce Schinier and Rivest also IBM but Belgium Algorithm was selected. However all the other algorithms are equally secure.

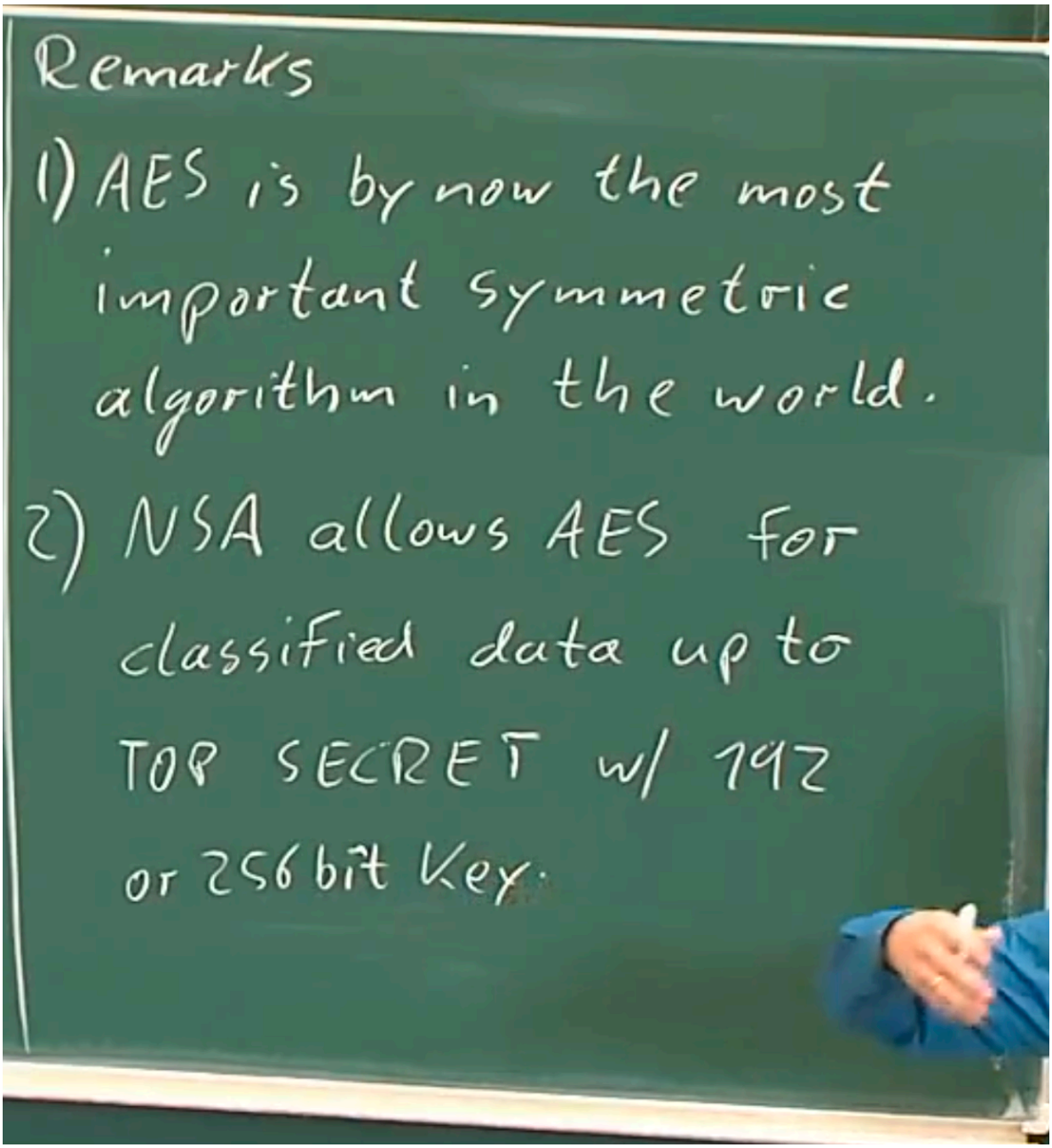
AES is a Block Cipher. As per specifications given by NIST, algorithm should be 128 bit block cipher (DES was 64 bit),

Plus number of rounds in AES depends on Key Length but incase of DES it is fixed 16 rounds.

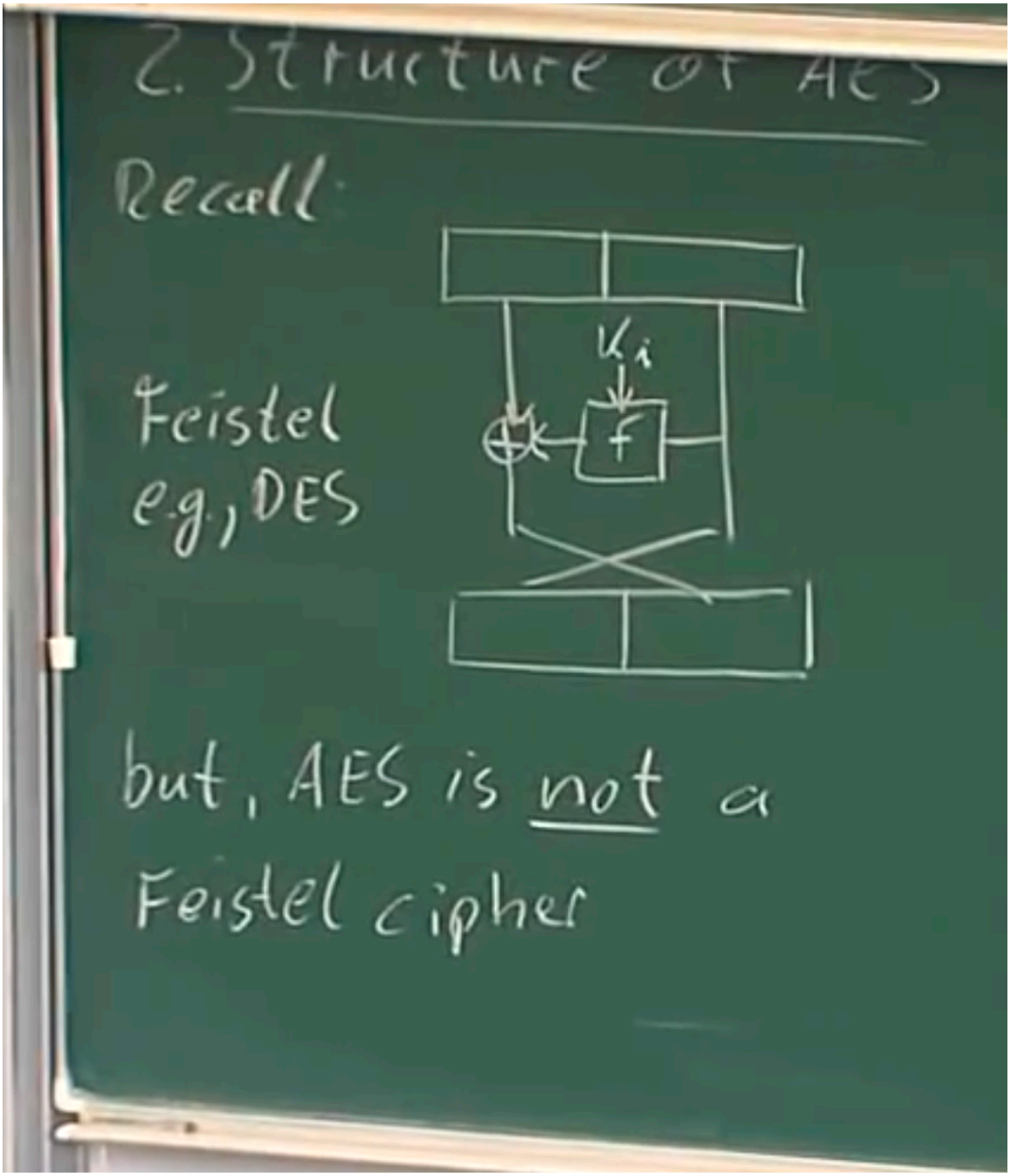
Key size for AES vary from 128,192,256.



For more security you need bigger key length. This was not in NIST specification.



Structure of AES :-
DES was a Feistel Cipher but AES is not a Fiestle Cipher.



In Feistel round only half of data path is encrypted (half old plain text). However AES encrypts entire 128 bit of data.

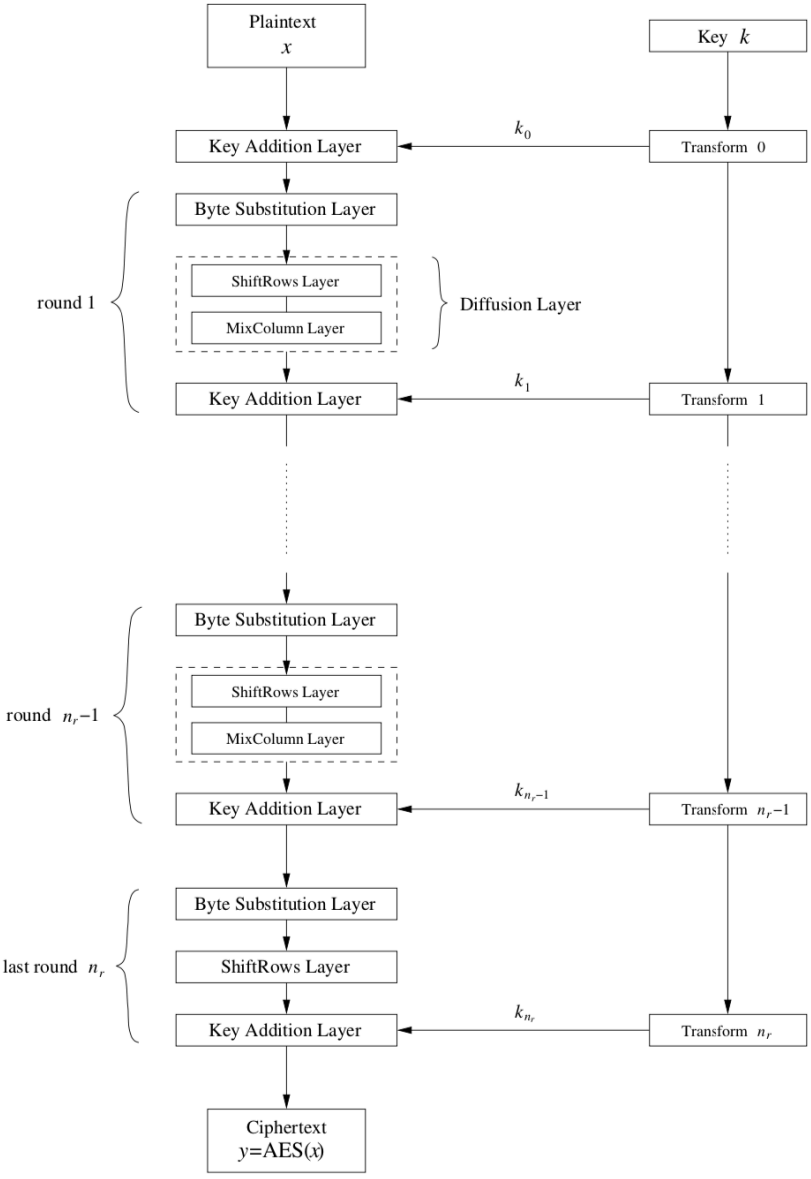
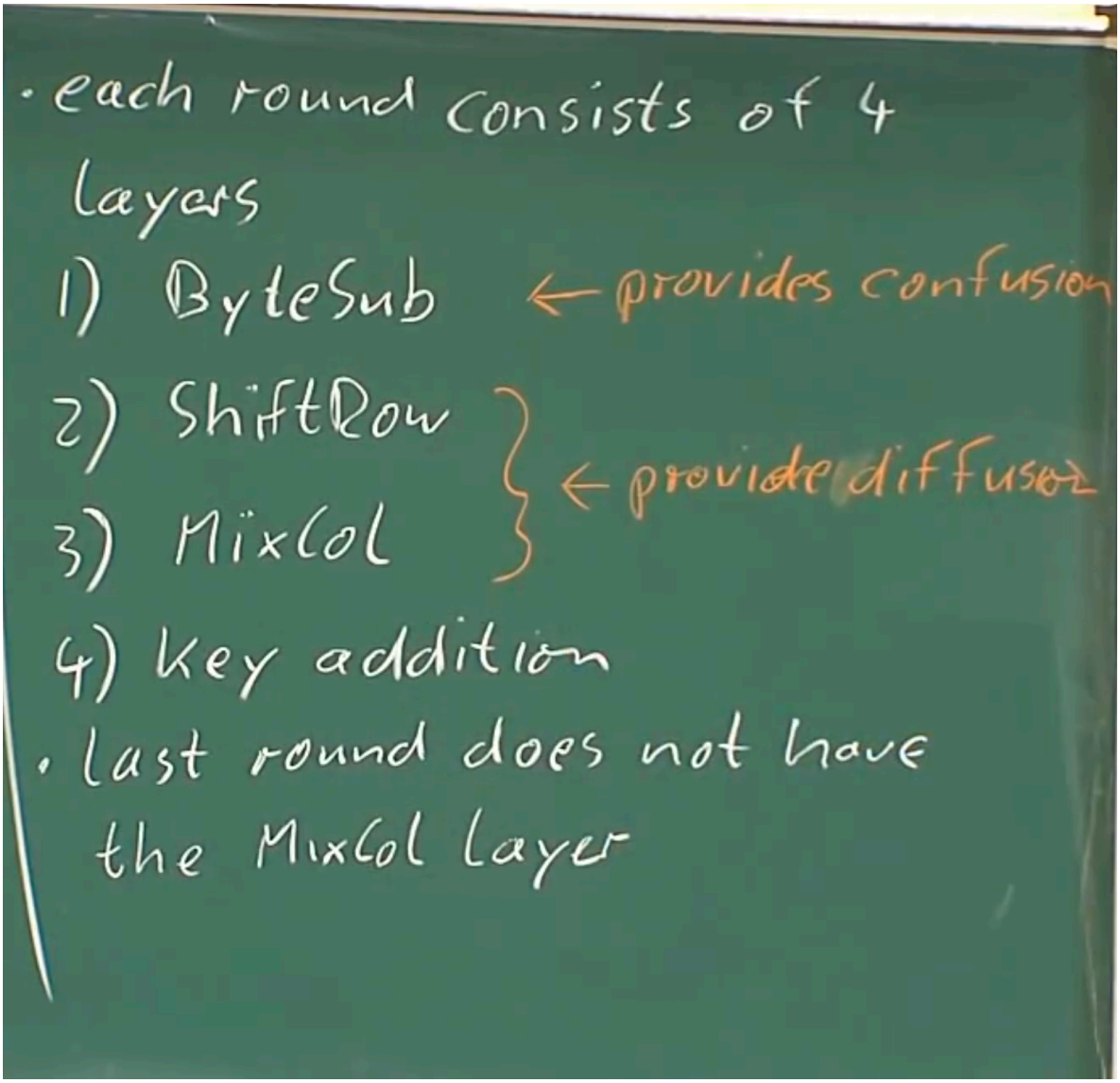


Fig. 4.2 AES encryption block diagram

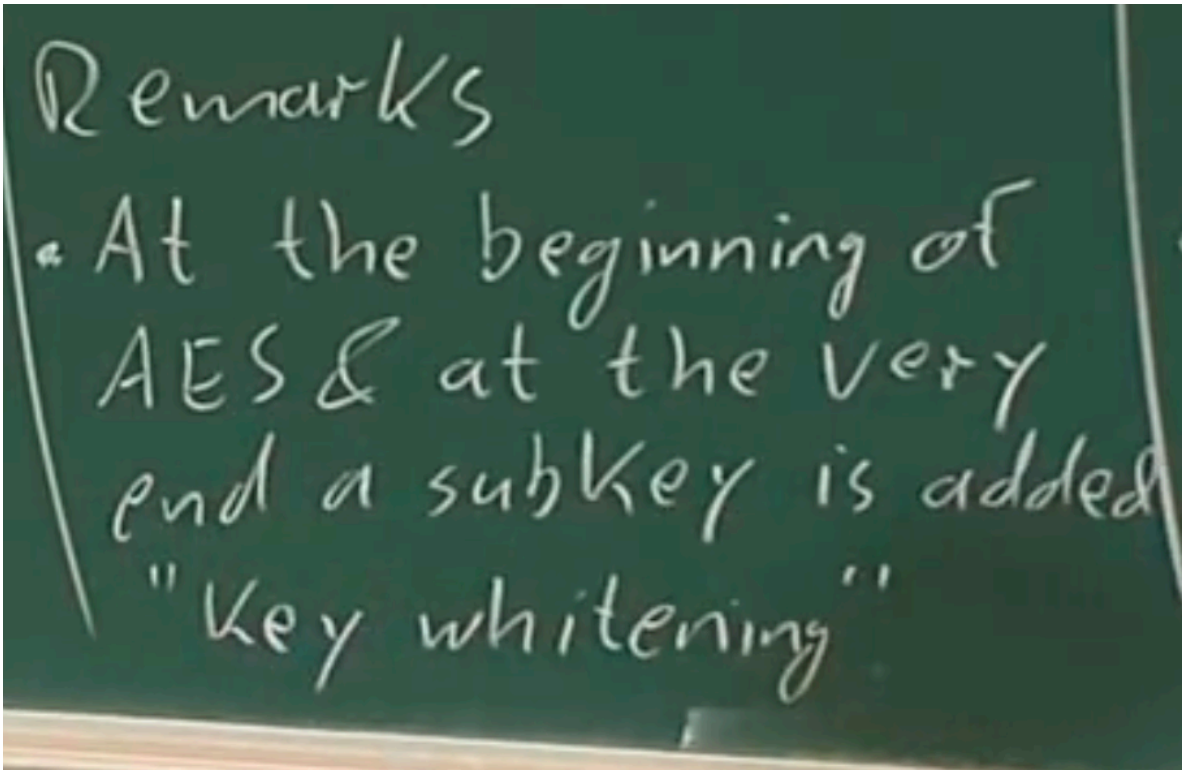
Remarks about the Figure 4.2

a. Each round consists of 4 layers :-
1. Byte Substitution Layer
2. Shift Row Layer
3. Mix Column Layer
4. Key Addition Layer

Also last round is little special as it doesn't have the Mix Column layer.



b. Key Addition :-
It was not there in DES but it is there for all the modern block ciphers. As the plain text comes in, first thing it does is adding SubKey and this is also done at the end. This is called Key Whitening.



Question :- What happens inside the layers ?
Internals of DES :- Fig 4.3

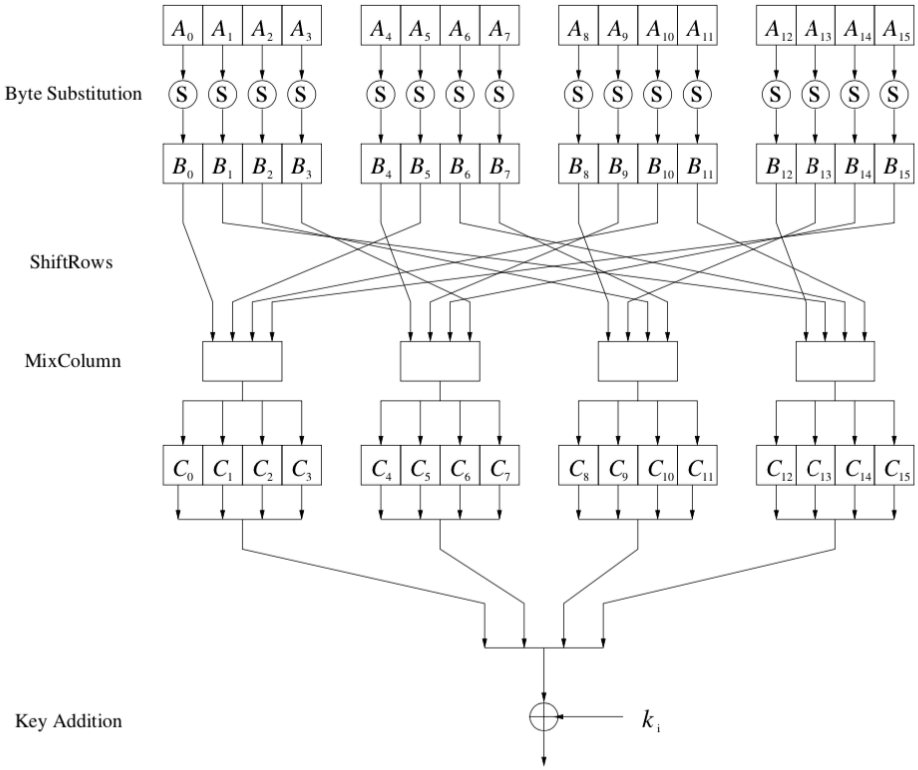
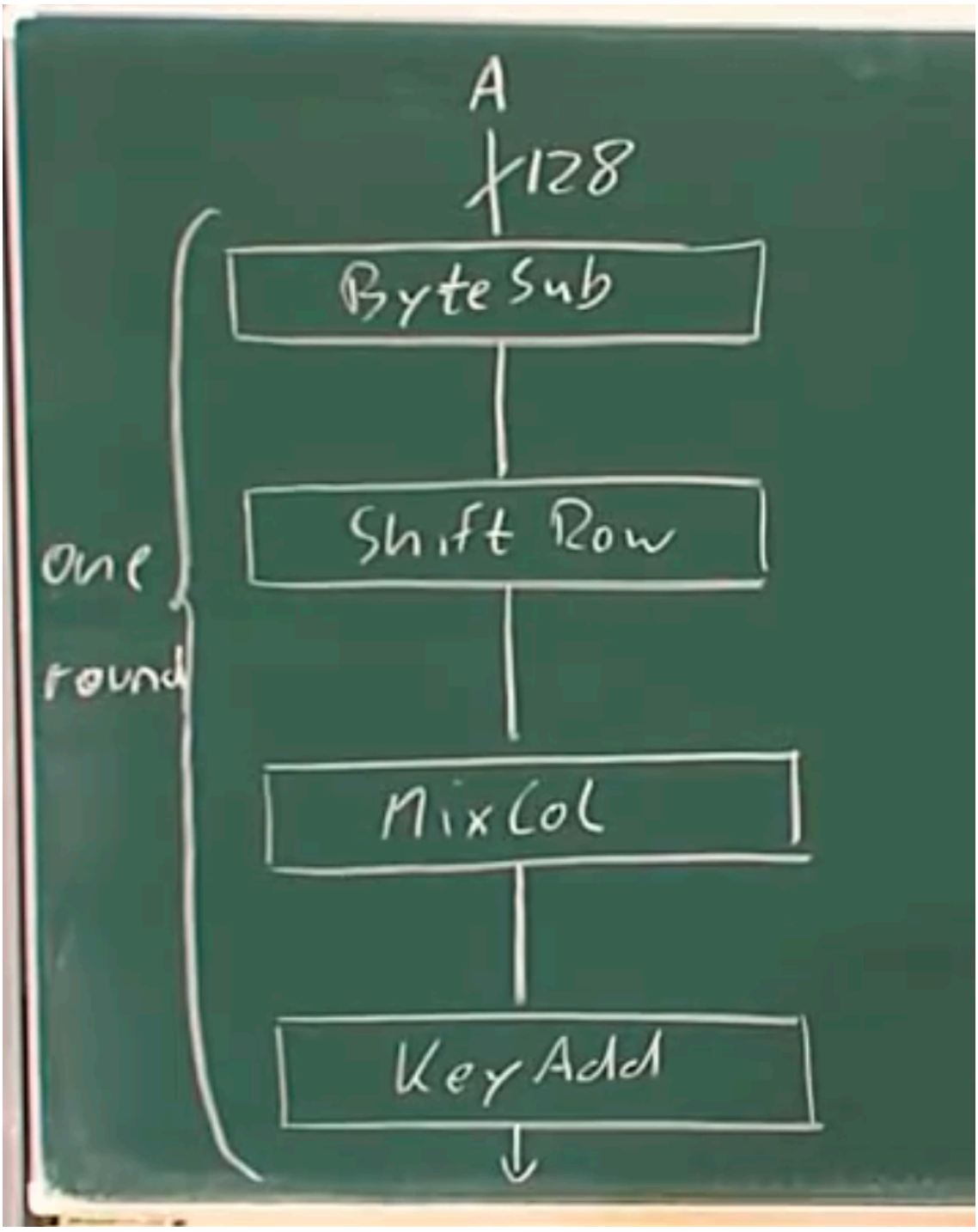


Fig. 4.3 AES round function for rounds $1, 2, \ldots, n_r - 1$

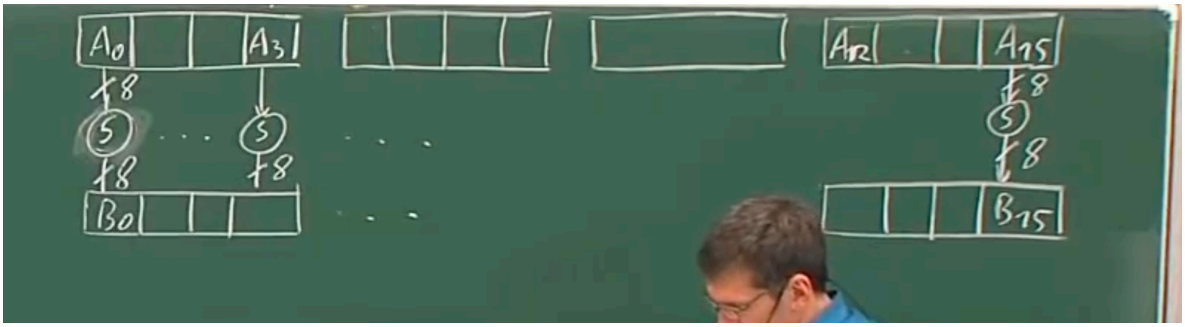
Note :- DES is bit based cipher where as AES everything happens to be Byte

based.In AES all operations deal with bytes ie 8 bits. DES is more bit oriented.

Byte Substitution is very similar in principle to what we do in DES.



A is 16 byte and we are grouping 4 bytes together.

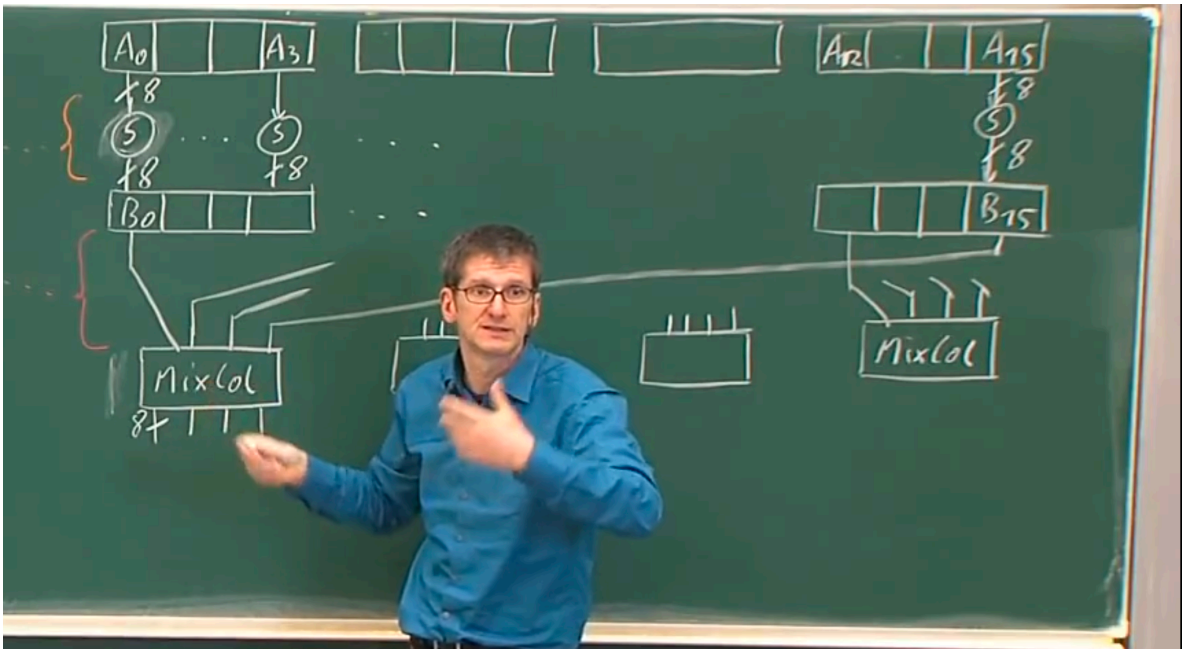


Here Byte substitution is done. In a Group of A ie A0 to A3, A4 to A7 … each 8 bit or 1 byte (A0 or any other) is substituted to B and something is done with Galois Field.

In DES we are always shifting/ permuting **bits** but in AES we are permuting Bytes in **ShiftRows** step.

One of the big feature of Modern Block ciphers is that we want to have diffusion at Algorithm Level.

Diffusion :-
Say you have one Plain text. X1 and it gives Y1 as output. Now changing one bit in X1 causing X1 to become X2, the assumption is Y2 should not look similar to Y1 ie say only one bit changed, this should not happen. So we want Y2 should look something like a Random number and doesn't have any similarity to Y1 (these should be no relationship that you could exploit as an Attacker.)



Experiment :-
Say we have A as the input (some bits). Now we flipped one bit in A0 byte so in Substitution, on an average 4 bits will change in B0.

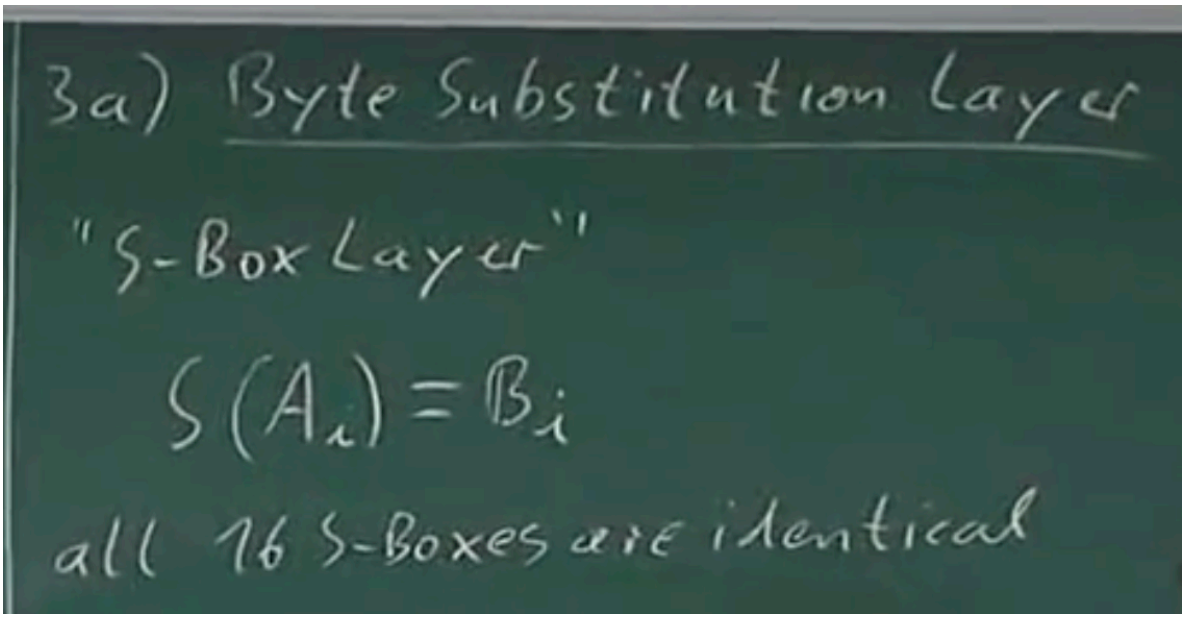
Now say we are going to MixColumn step where only one byte from A0 to A3 is added (A0) and all other bytes are coming from other groups. Now MixColumn will mix the bits with the entire group of ie all 32 bits/4 bytes.
So point is you have affected one bit in Plain Text, in one round all 32 bits are affected ie one out of 4 groups is affected with one bit changes.

Some explanation :-
When we do byte Substitution we are just changing bytes in one block(local) ie A0 or A1 etc. Also in Shift Row we are shifting the byte blocks but we want to affect other byte blocks too. So for that we need to do **Diffusion.**

So AES is having a very strong diffusion.

Mix Column is Matrix Multiplication.
In key addition, we are going to XOR with the SubKey of each Round.

Byte Substitution :-
It is Substitution layer (S-Box). Difference between AES and DES S-Box is AES has identical S Boxes but DES is having 8 different S boxes.



[Table 4.3]

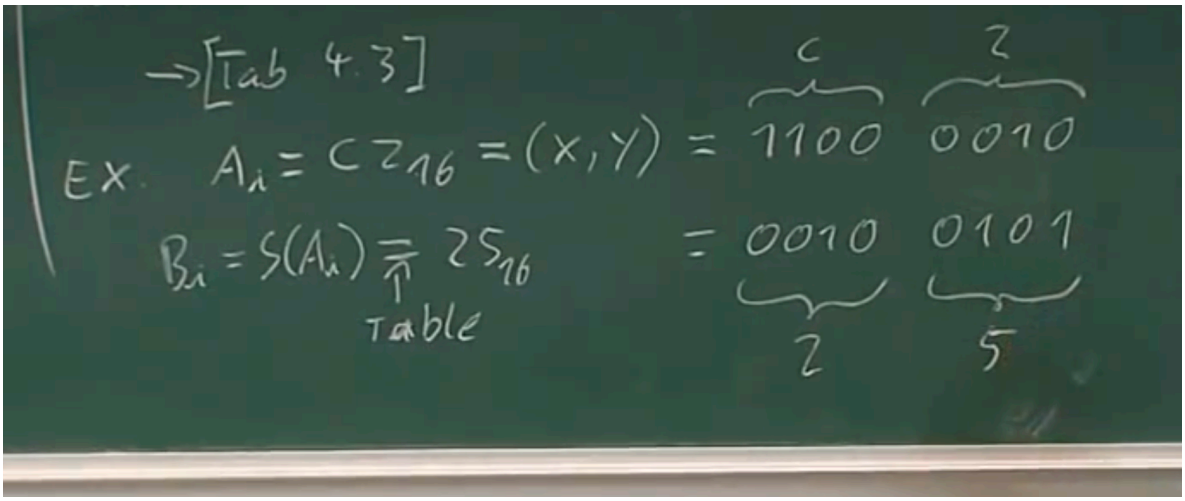
Table 4.3 AES S-Box: Substitution values in hexadecimal notation for input byte (xy)

	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
x 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

Example 4.8. Let's assume the input byte to the S-Box is $A_i = (C2)_{hex}$, then the substituted value is

$$S((C2)_{hex}) = (25)_{hex}.$$

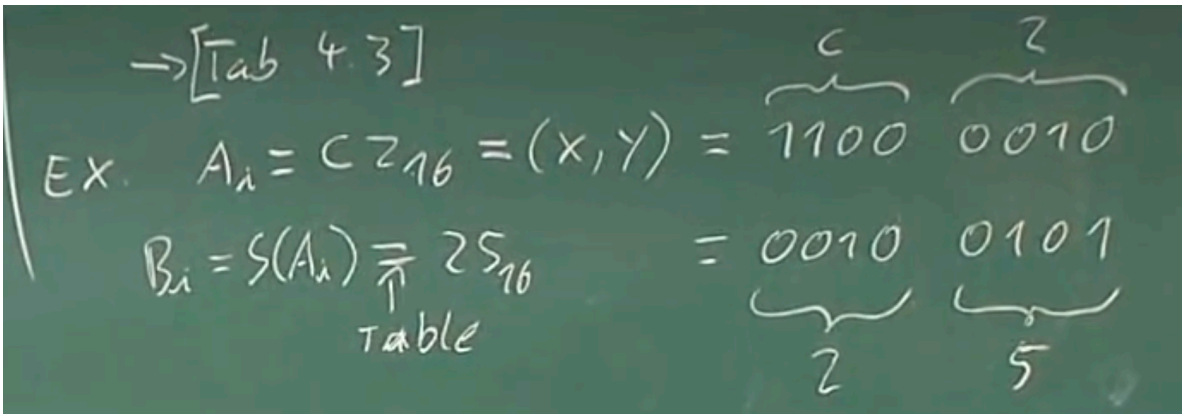
Notation to read the table :-



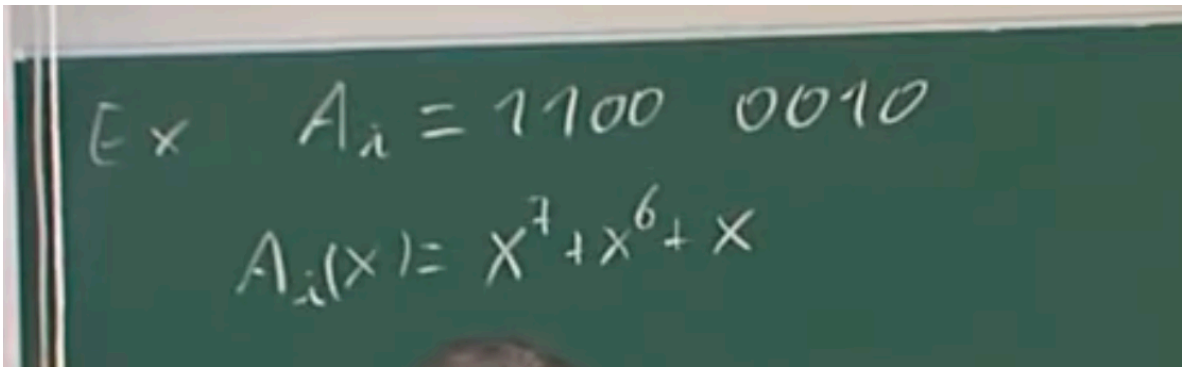
First Digit tells about the Row and Second tells about the column.

Question : How is the S-Box table constructed ?
In DES we are not sure how the S-Boxes are constructed but we can know how AES s-box are constructed. In DES they randomly chose S-Box and then checked if requirements are fulfilled.

It is important to note that these AES S-Boxes are constructed in a very specific way.
Point is Input can be considered as a bit vector if you look in below table :-



So Here comes the Galois Field. Galois field is used for computing S Boxes. Now you can see a bit vector and it can be a polynomial in GF.



Description from Book:
S Box computation is a combination of 2 steps one is finding inverse of GF and other is affine mapping.

step mathematical transformation (Fig. 4.4):

As we know inverse of 0 doesn't exists so AES map 0 to itself.

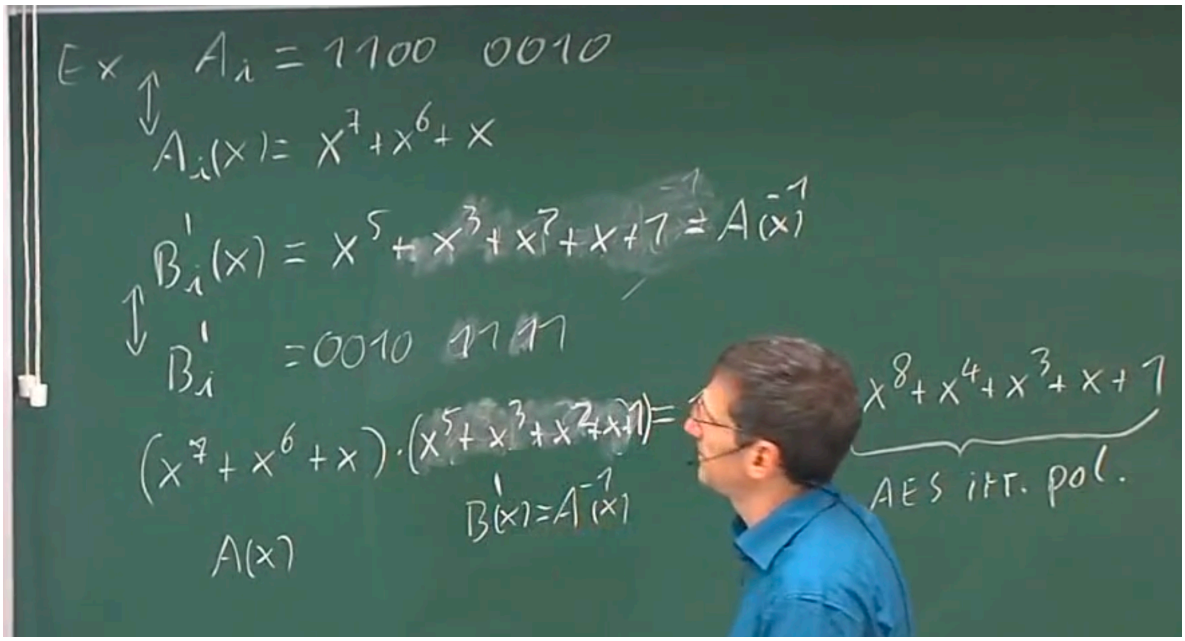
Fixed irreducible polynomial for AES $P(x) = x8 + x4 + x3 + x + 1$

Affine Mapping:

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} \equiv \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \mod 2.$$

t $B' = (b'$ $b')$ is the bitwise vector representation of $B'(x) =$

above is the way how the fixed Constant vector matrix is multiplied and a constant vector is added to generate the final S Box.



Question is why are we doing Inverse of GF ?
Answer: because this gives us very good cryptographic properties. GF inverse provides a higher degree of Non Linearity.

Affine step destroys algebraic structure of GF to prevent the attacks which exploits the Inverse of Galois Field.

Question, you might think of computing GF inverse on every encryption but actually this is not needed as it is done once and we can use that S box in the Byte Substitution.

However in Hardware it is sometimes little tough to store these small tables so in hardware we generally compute these GF inversions.

Shift Rows:
While looking at the below image we might think Shift Rows is random like DES but actually it is very Systematic.

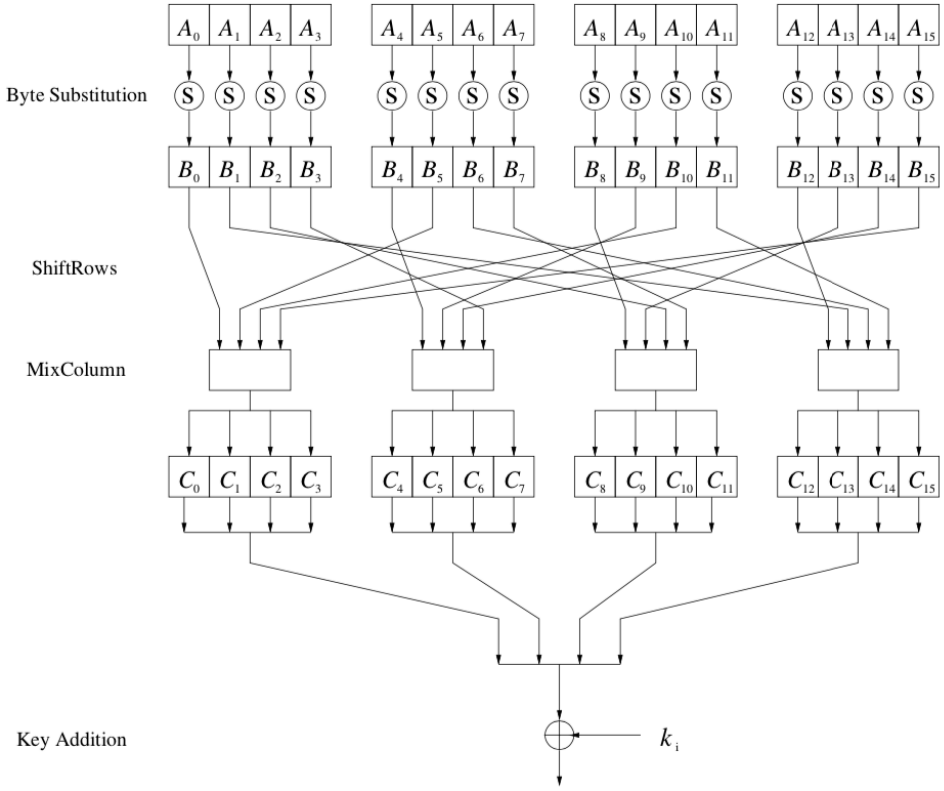
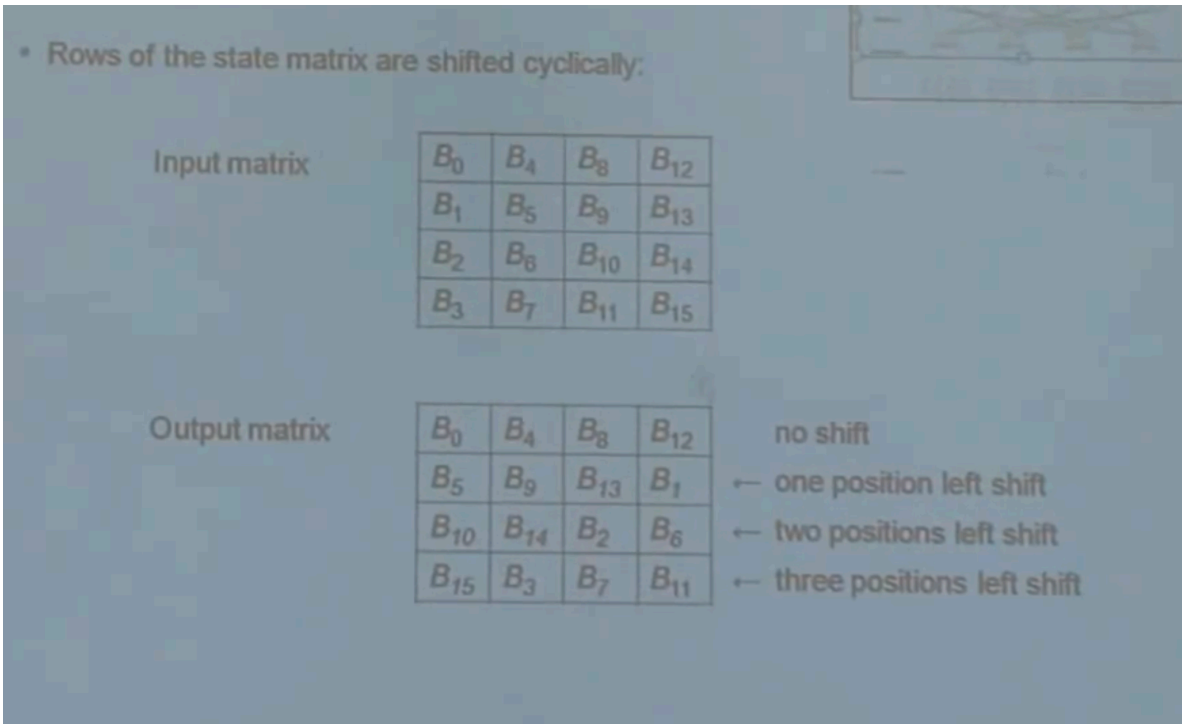


Fig. 4.3 AES round function for rounds $1, 2, \ldots, n_r - 1$

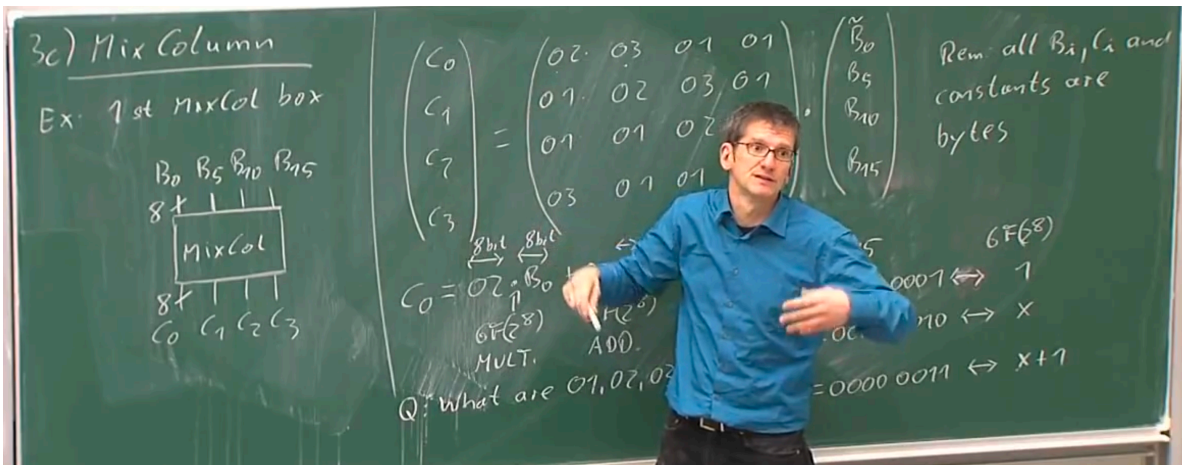
if you look at these 16 bytes as a Matrix then following is the way how Shift Rows is done :

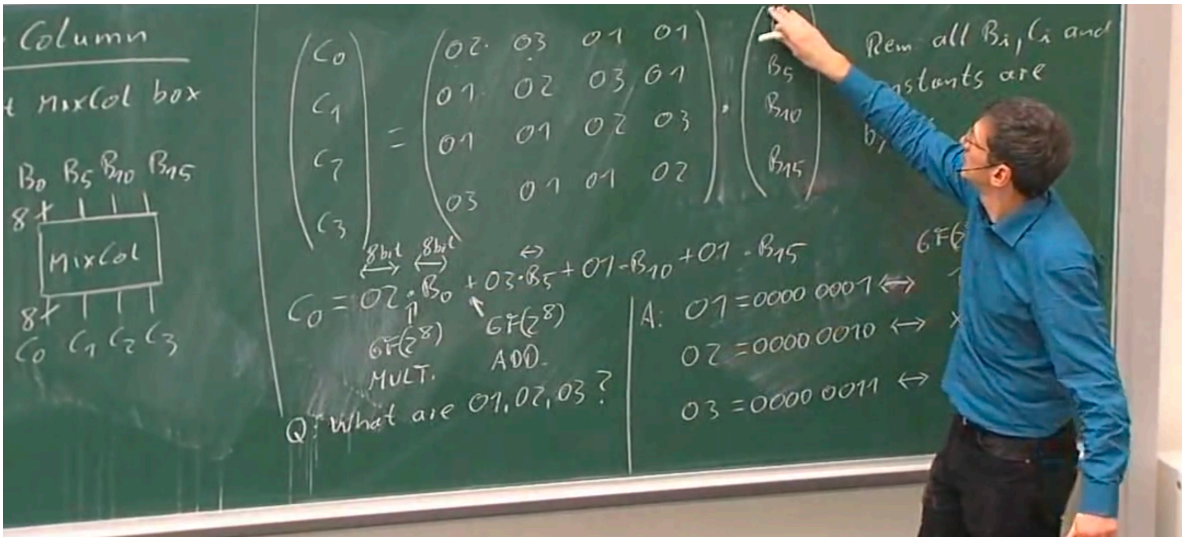


if you look carefully each byte group is connected to all the mix column.

Mix Column:

So say we have changed one bit in plain text so Byte Substitution might change the 4 bits (on a average) and those 4 bits are changed for the same byte block so total 8 bits are effected. now we want to increase this effect in Mix Column to 32 bits. so what we need to do to achieve this ?



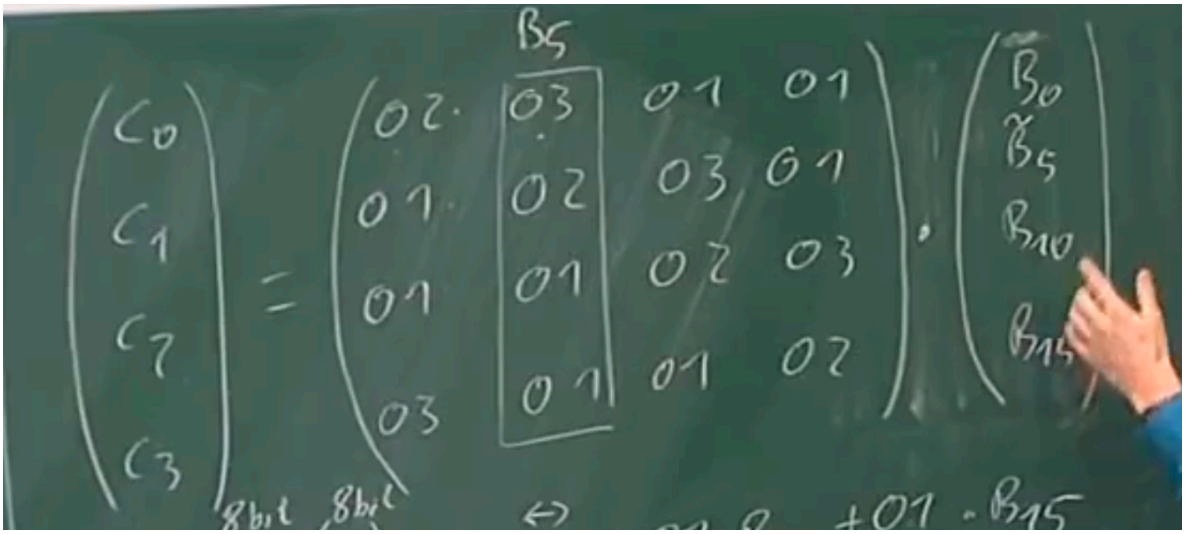


so what is happening in Mix Column is matrix multiplication. so there is a constant Matrix

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{pmatrix} \begin{pmatrix} B_0 \\ B_5 \\ B_{10} \\ B_{15} \end{pmatrix}.$$

which is multiplied with the input bytes of the Mix Column and question is how the multiplication is done. so multiplication is done using Galois Field multiplication and Modulo reduction with the same vector which we used in S-Box computation (Fix AES irreducible polynomial) Also Modulo addition also happens in GF.

Now question is why are we doing this ? as you can see C0 is going to be computated using all the 4 B0, B5, B10, B15 so on a single bit flip, every C0,C1,C2,C3 is going to be effected.
also here 01 is represented in a polynomial as 1 and 02 as X and 03 as X+1.



left part in AES are Key Schedule and Decryption.