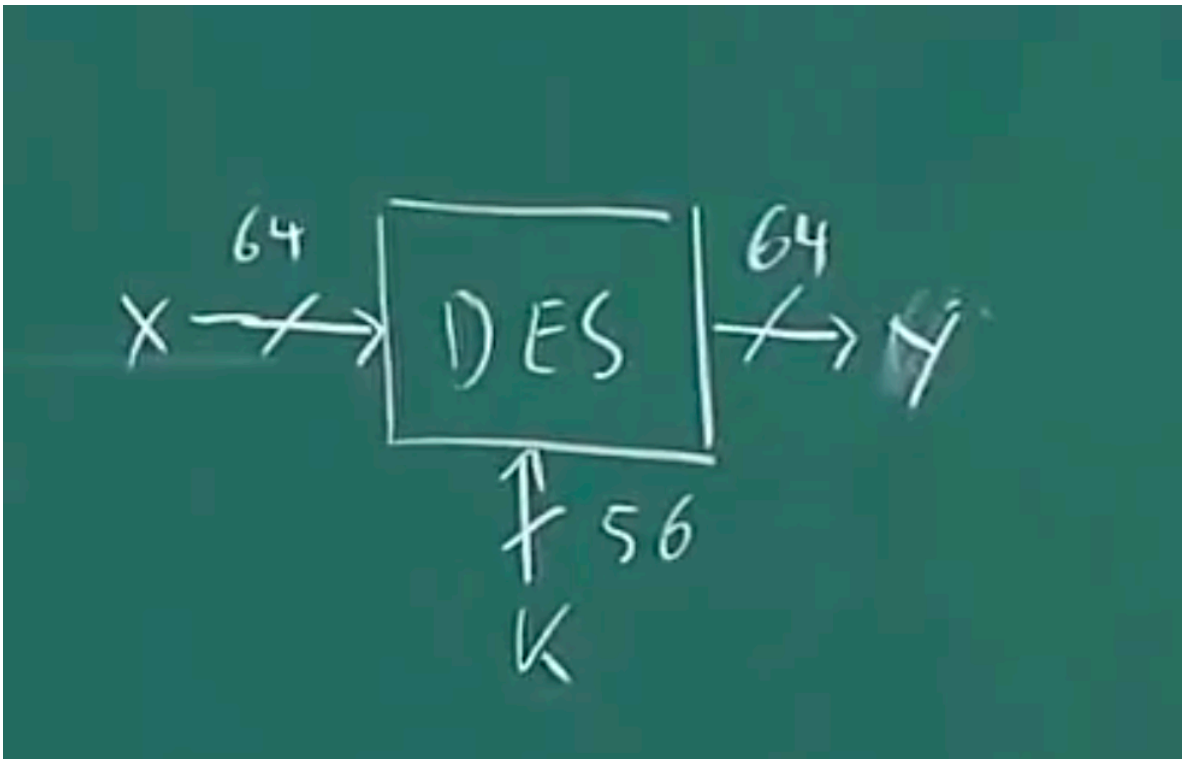
**Lecture 5 Data Encryption Standard :-**

- Developed by IBM with inputs from NSA (US).
- US government has made it as standard for applications.
- Most Studied cipher
- Unsecure today due to Key too short.
- Triple DES is very Secure. It is something similar to 3 times DES encryption.

**High Level design of DES :-**
DES :- Block length is 64 bits and Key length is 56 bits.
It is not stream cipher where we encrypt individual bits but here we encrypt the block of data.




**Question :- What is the good way of building Block Cipher ? Or How do we build the Block Cipher ?**
— Building Cipher is an Art than Science. — Cristof paar.

**Shannon :-** has done majority of work in information theory and he has also done some work in cryptography.
He told that building the block ciphers is pretty difficult but there are 2 general principles :-

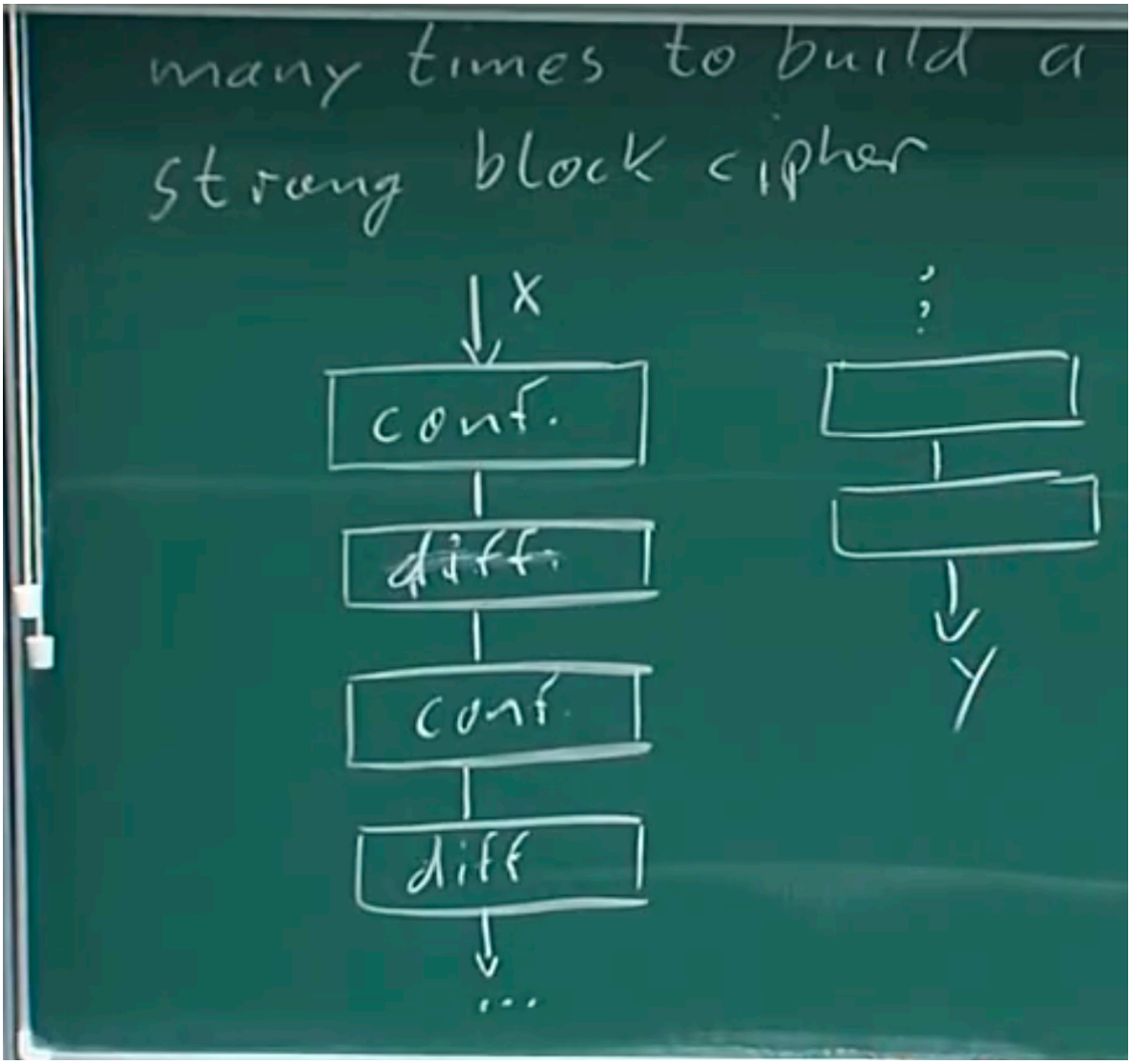
**Two atomic operations :-**

1. Confusion
2. Diffusion.

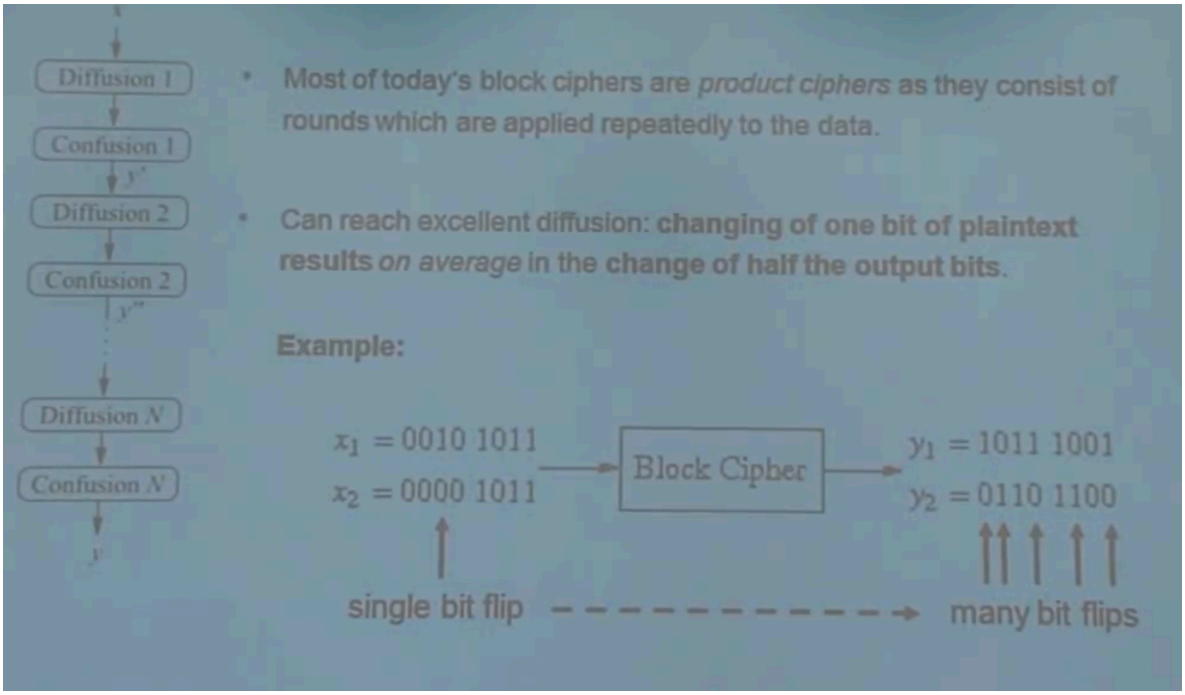
1. **Confusion** :- Relationship between plaintext and cipher text Is Obscured.
   eg :- Substitution table.
2. **Diffusion** :- only Confusion is not enough, you need diffusion ie the influence of each plaintext bit is spread over many Cipher Text bits. Eg Permutation.

— Combine Confusion and Diffusion many times to build a strong block cipher.

All the Ciphers we used are Confusion but we should do Diffusion.



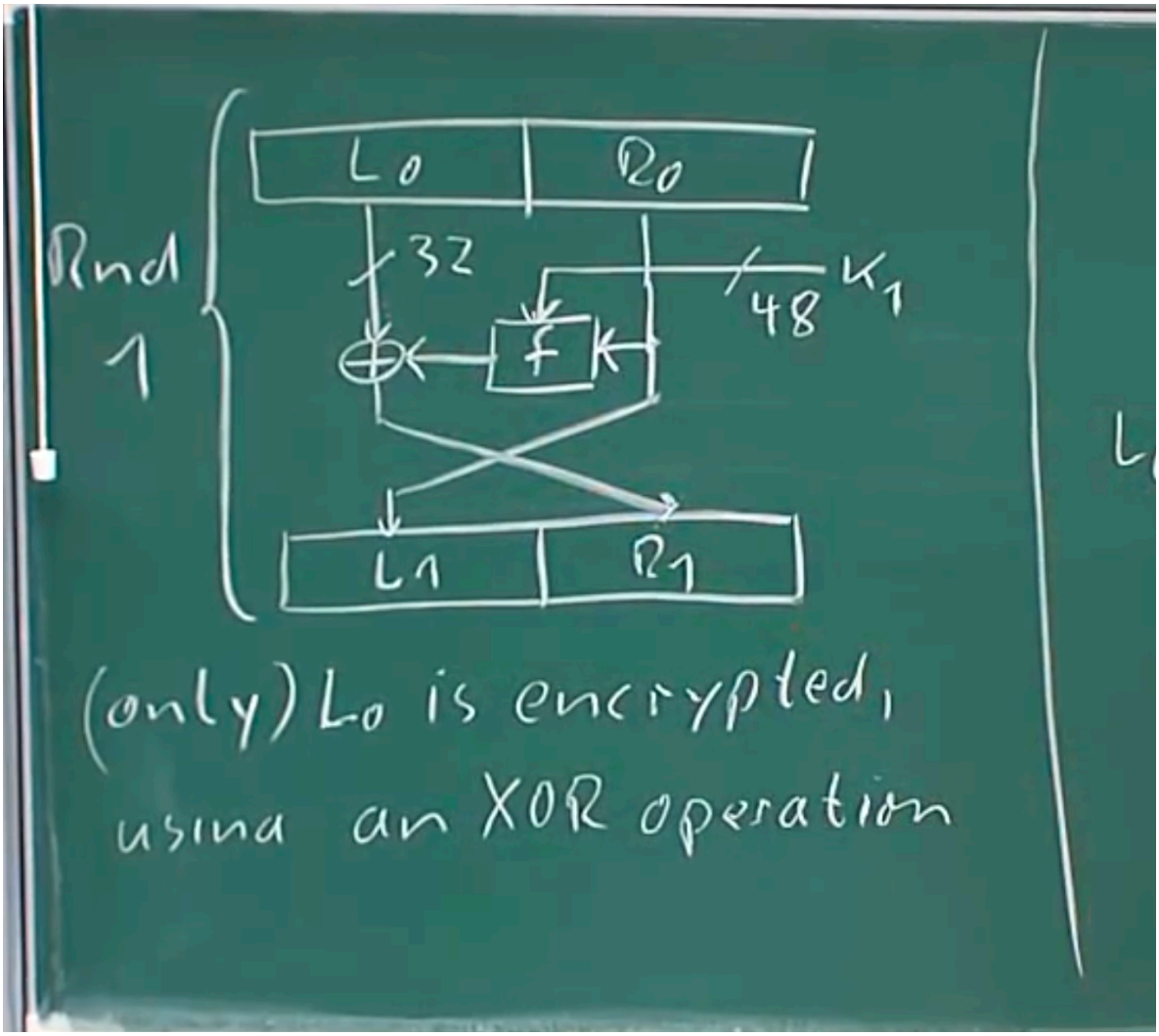

This is called “**Product Cipher**”.

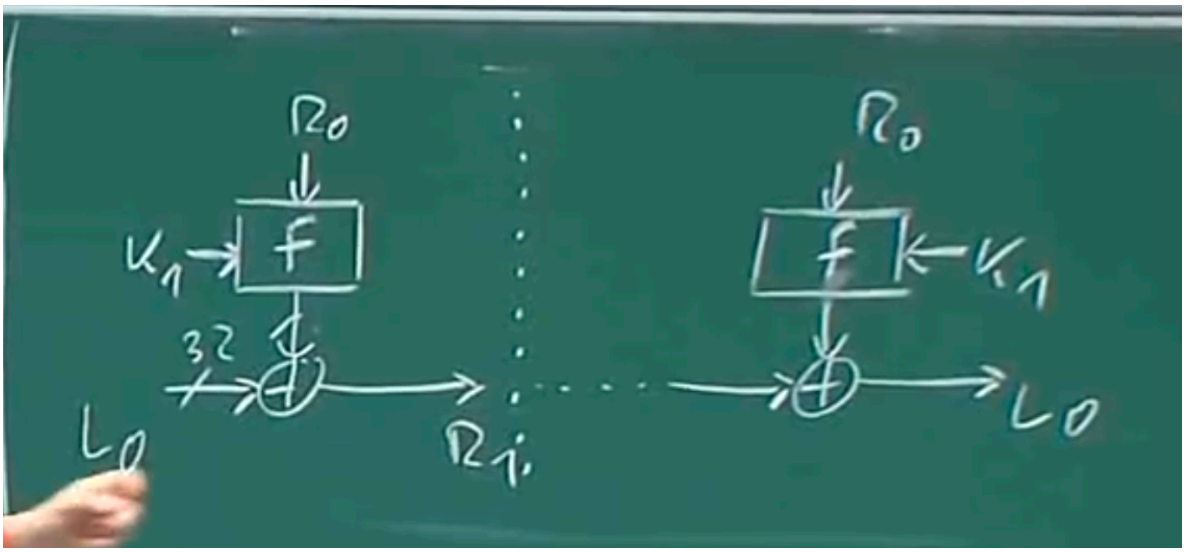



**Feistel Network :-,**

Many of todays ciphers are Feistel Ciphers (but not all !!!)
DES does Confusion and Diffusion 16 times ie 16 rounds.

**Question** is What DES does in each round ?




Here while decrypting how can we know R0 ? Actually if we look at feistel network ie one above this picture, we can notice that R0 is same as L1 as we have a no-op there.

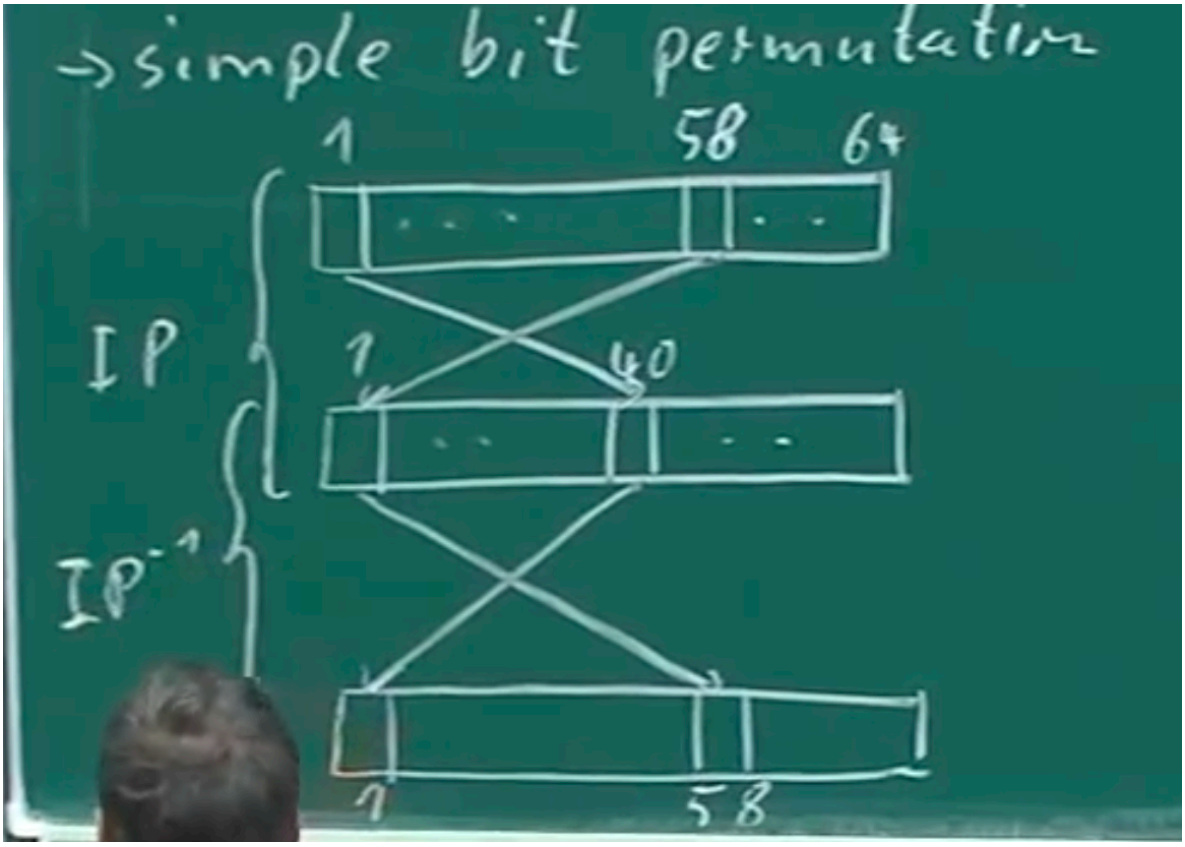
So incase you think L1 is not needed then you cannot decrypt the message.

Question ? What next we are left with in DES ?
Details ie F function :-

3. **Internals of DES**

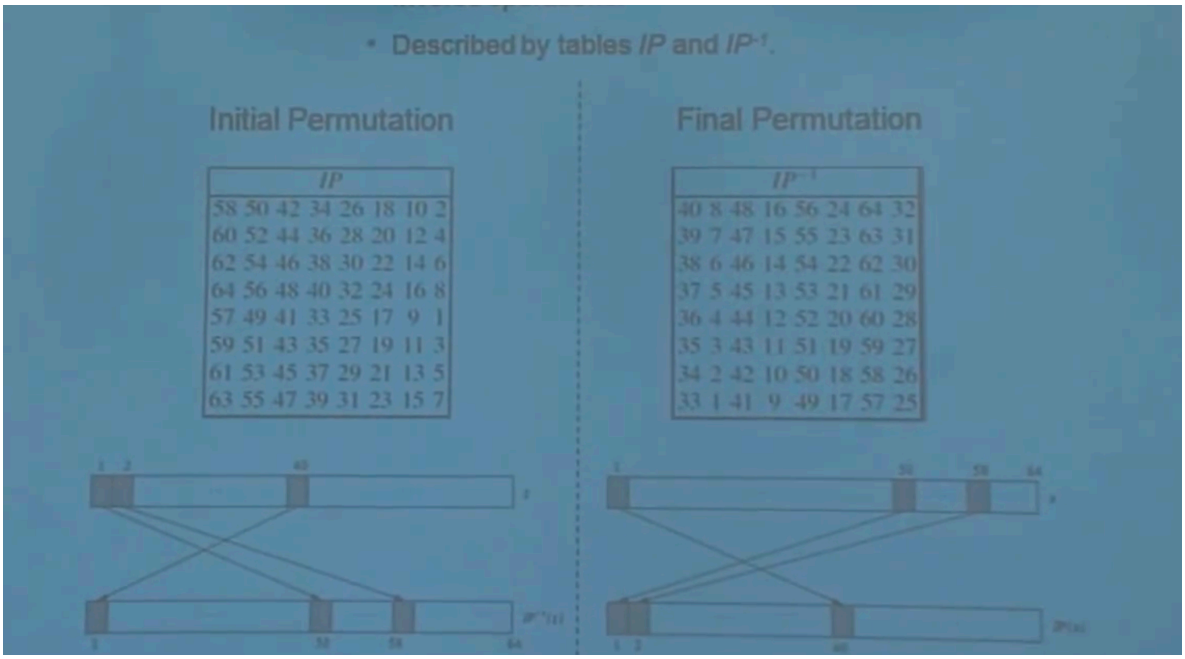
**3a** -> IP and IP inverse
First step in DES while encryption is Internal Permutation not Round Function.
In bottom/last we will do IP inverse.

Simple Bit permutation :-



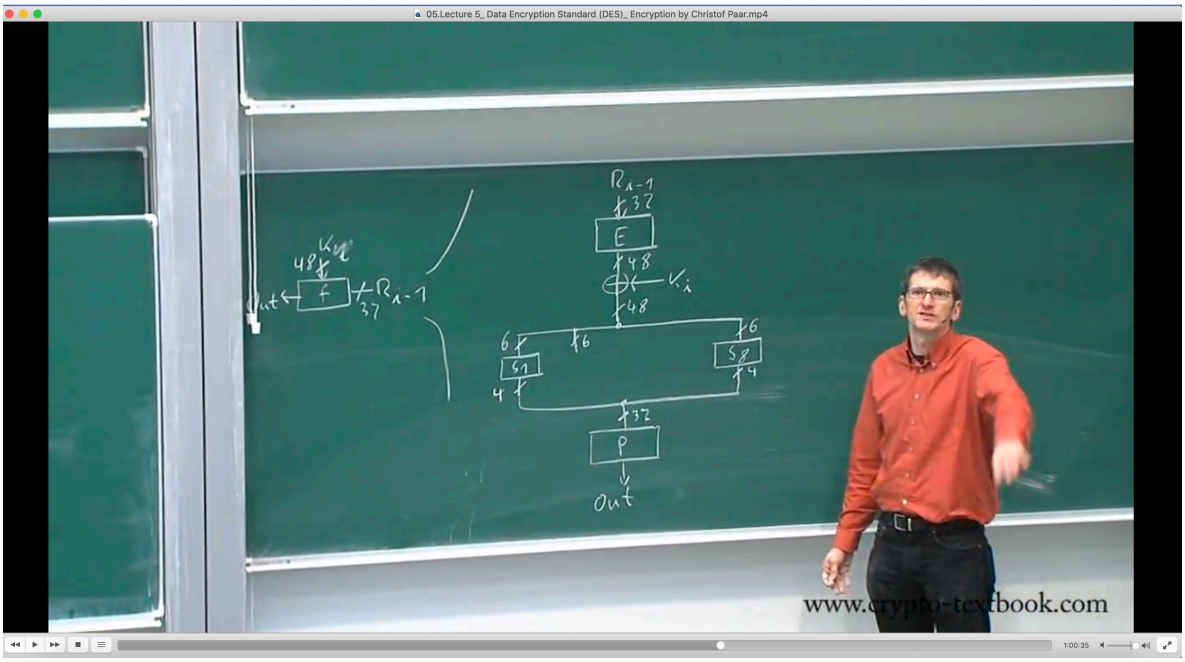

If we do an IP and then inverse IP we will get the original plaintext.
**Question arises why are we doing Initial Permutation ?**
As IP table is publicly known then it doesn't makes sense, it doesn't add security.
This was because of some electrical engineering problems (Implementation issues) and it should not be part of specification but some how it became.

There was one more theory, that DES is only allowed in Hardware as per NSA so people thought that may be IP operation is very easy and it doesn't cost anything something like a free. In Software it might have little complexity.
Above theory is not correct.

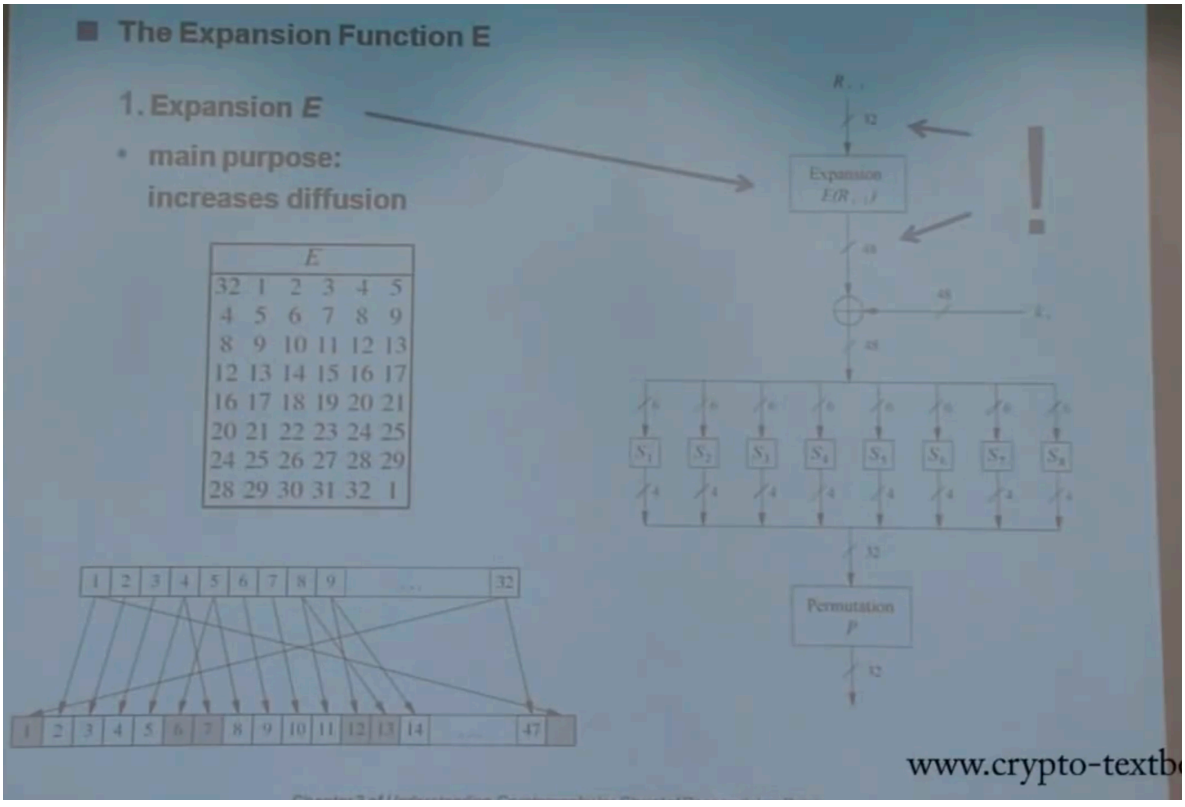



Notation of above table is 58th position in input is 1st position in output.
**3b ->** Details of F function :-
Inside F function main 4 operations we are doing.

1. Expansion Box -> Input to expansion box is 32 bit and output is 48 bit. I think they have to add the expansion box as Key size is bigger than the 1/2 Block size as feistel divides block into 2 parts and only operate on one part. Subkey size is 48 bit.
2. Then we have 8 S box —> Substitution Box where 6 bits are substituted by 4 bits.
3. After the combining S box output we get 32 bit as output.
4. Then we do permutation.

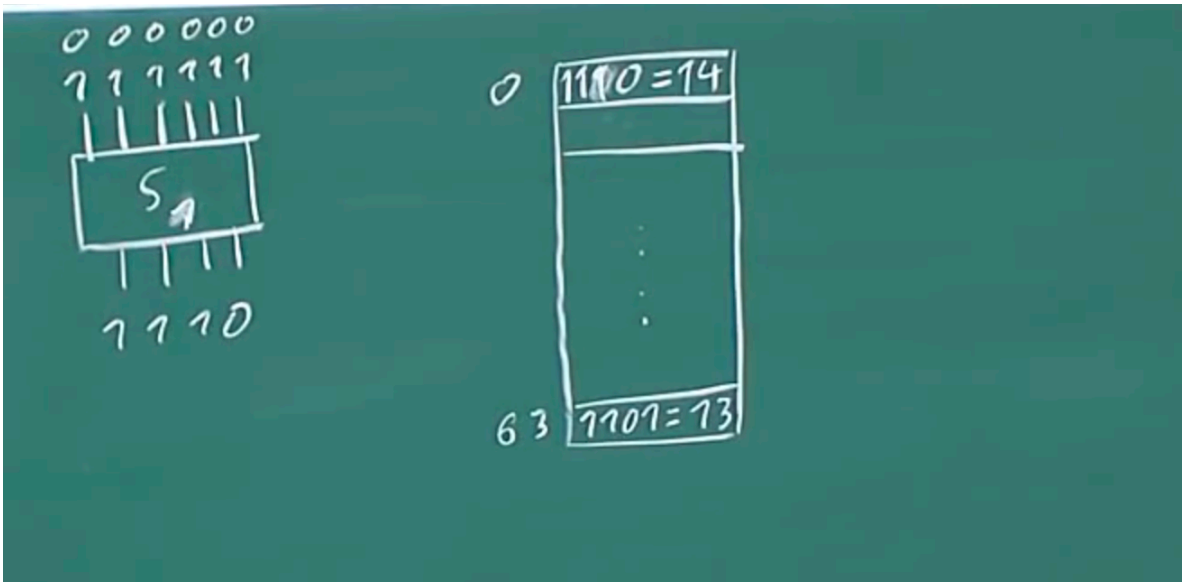



**Expansion Box :-** provides diffusion. We are going from 32 to 48 bits now question is why we are even doing expansion ?
We get diffusion.




16 bits have 2 connection and 16 have one connection.
As we have seen, change in one bit plaintext impacts multiple positions in Cipher Text.
If you see expansion, you might notice that it is just impacting 2 bits but going further you will see it makes a bit difference in DES.

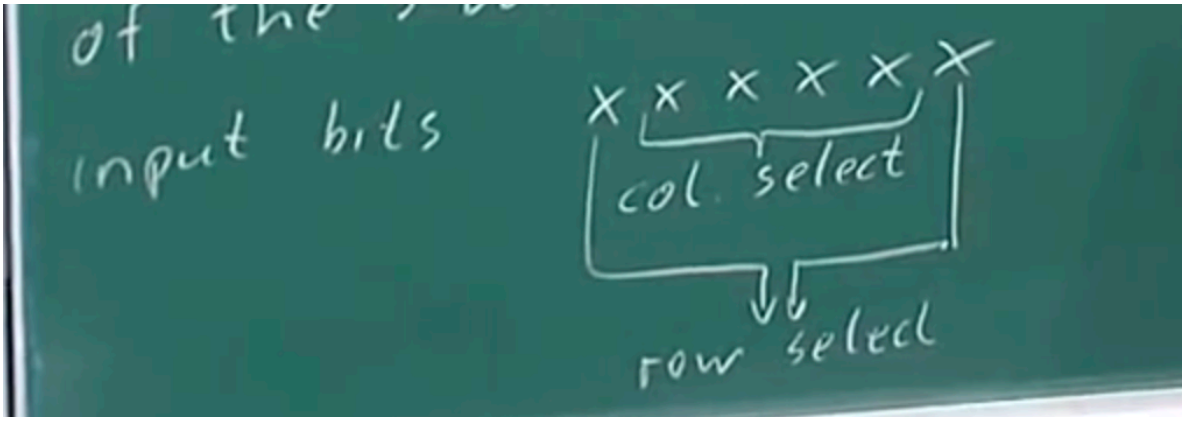
**S-Box :- Heart of DES -> Provides Confusion.**
S box converts 6 bit input to 4 bit input.




| $S_1$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 14 | 04 | 13 | 01 | 02 | 15 | 11 | 08 | 03 | 10 | 06 | 12 | 05 | 09 | 00 | 07 |
| 1 | 00 | 15 | 07 | 04 | 14 | 02 | 13 | 01 | 10 | 06 | 12 | 11 | 09 | 05 | 03 | 08 |
| 2 | 04 | 01 | 14 | 08 | 13 | 06 | 02 | 11 | 15 | 12 | 09 | 07 | 03 | 10 | 05 | 00 |
| 3 | 15 | 12 | 08 | 02 | 04 | 09 | 01 | 07 | 05 | 11 | 03 | 14 | 10 | 00 | 06 | 13 |

In Slides It is mentioned it is good to protect against differential cryptanalysis.

If you look at table, it is different as it has 16 columns and 4 rows but how we can read it as we have input of six bits.
**Notation of reading this table is 4 bits choose column and 2 bits choose rows.**




Middle bits are used for column select and outer bits are used for row select.

When IBM.and NSA were telling about the DES, people are able to understand very quickly but they are not able to understand from where these S boxes are

coming and why out of so many combinations of numbers only this table or similar tables were chosen ?
Actually all the tables S1 — S8 all look similar and looks pretty random.
**IBM told that it is secure trust us. But people are thinking that there is some backdoor etc.**

Two Israel scientists tried to break DES and they were successful using differential cryptanalysis. Differential cryptanalysis depends on how the structure of s-box ie how tables are filled. The author of DES told that they know this 15 years back.
When they went to practically break DES, it doesn't work because S-Box are particularly chosen so that attack is not successful.

**Experiment** :-
Say we have 32 bits all Zeros in F function, you got a certain output now say we have flipped one bit.
Question :- What happens in Expansion box ?
Ans :- two things can happen, output bits will flip either one bit or 2 bits. 50-50 chances.
Taking one simple case of 1 bit flip.
Now XOR :- now we know that output will flip one bit no matter what the key is. (Key remains same)

Now comes interesting property of S-Box :- if one input bit flips, atleast 2 output bits flips.
P box will try to put these bits geometrically farther ie they don't say closer.
Now say in next round, you have 2 bits different and now chances are that in expansion box we meet more that 2 bits (ie 2 bits for one of the changes bit) but say due to bad luck you found only one output connection.

Due to permutation of previous F function, say one bit goes to S1 and other goes to S5. Now we can say at least 4 bits are different now. And so on....