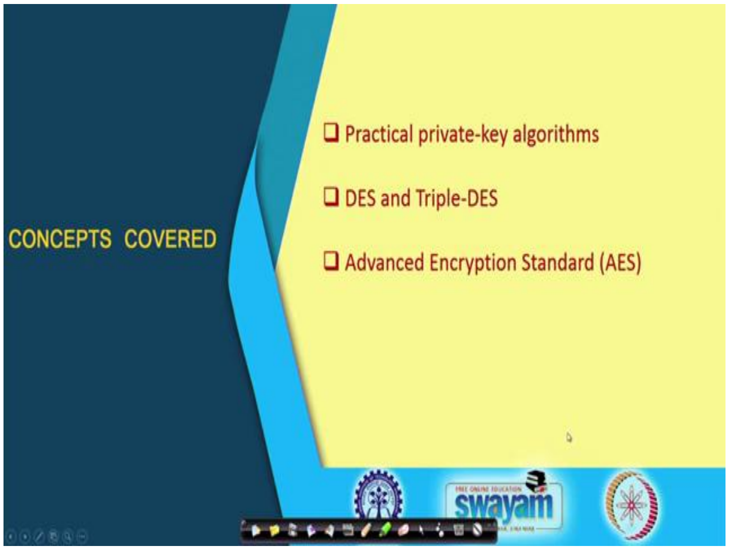
Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 28
Private - Key Cryptography (Part II)

Continuing with our discussion on Private-Key Cryptography, in this lecture we shall be looking at some of the practical encryption/decryption algorithms which have been used or which are being used. So, the topic of this lecture is private key cryptography the second part.

(Refer Slide Time: 00:35)

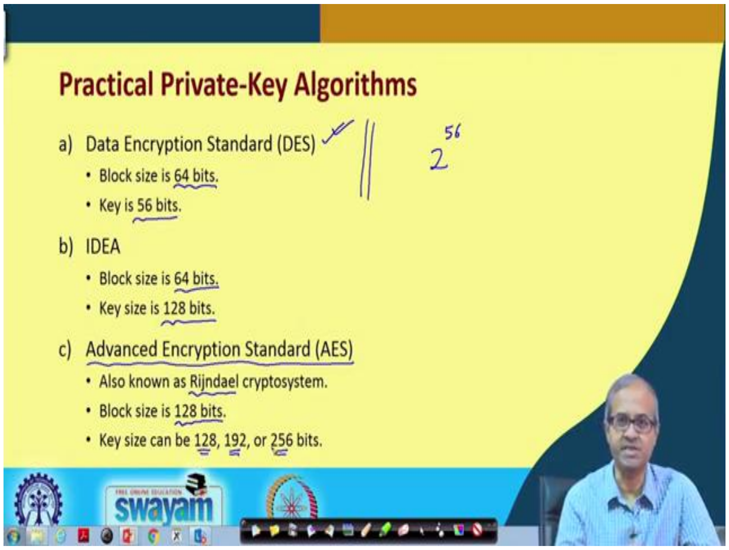


CONCEPTS COVERED

- □ Practical private-key algorithms
- □ DES and Triple-DES
- □ Advanced Encryption Standard (AES)

Now, as I said now here we shall be revisiting some of the practical algorithms, which are used in real scenarios. The algorithms that we will be talking about our Data Encryption Standard or DES, triple DES which is an extension of that and the most recently one, recent one which is most widely used nowadays is the Advanced Encryption Standard or AES, ok.

(Refer Slide Time: 01:01)



Practical Private-Key Algorithms

a) Data Encryption Standard (DES) ✓ 2^{56}
- Block size is 64 bits.
- Key is 56 bits.

b) IDEA
- Block size is 64 bits.
- Key size is 128 bits.

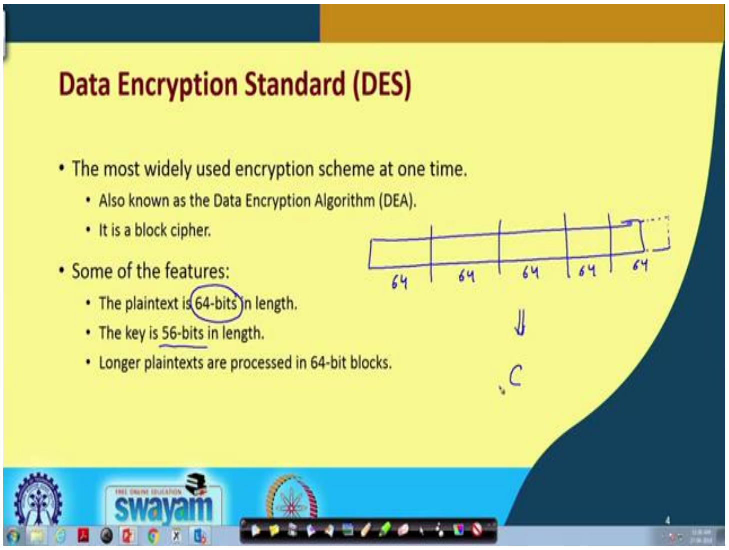
c) Advanced Encryption Standard (AES)
- Also known as Rijndael cryptosystem.
- Block size is 128 bits.
- Key size can be 128, 192, or 256 bits.

Talking about the practical algorithms, there are numerous algorithms in fact. If you look at the literature, in the literature there is lot of research going on in the development of new algorithms. There are good features in these algorithms, there are weaknesses, there are drawbacks, but the cryptographers or the persons who are developing these algorithms they are continuously trying to come up with better and better algorithms. So, data encryption standard was one of the algorithms, which was proposed earlier in the 80s, where the block size was 64 bits.

So, it encrypted 64 bits of data at a time and the key size was 2, was 56 bits. This DES was used for quite a significant amount of time, but because the key size is not that large, 2 to the power 56 in the present day context is not a very large number. Using the fastest computer with brute force technique, you can generate all these keys and you can mount a brute force attack. So, there are other algorithms which have been proposed, this idea is one of the algorithms which have been explored here also the block size is 64 bit, but the key size has been enhanced to 128 bits.

But as I said the most widely used symmetric key or private key algorithm today is called advanced encryption standard or AES. This is also referred to as Rijndael cryptosystem taking a cue from the names of the inventors. Here the block size is 128, but for the key size we have a choice. You can have 128, 192 or 256 bits depending on the level of security that you want in a particular application; larger the key size, higher will be the security.

(Refer Slide Time: 03:19)

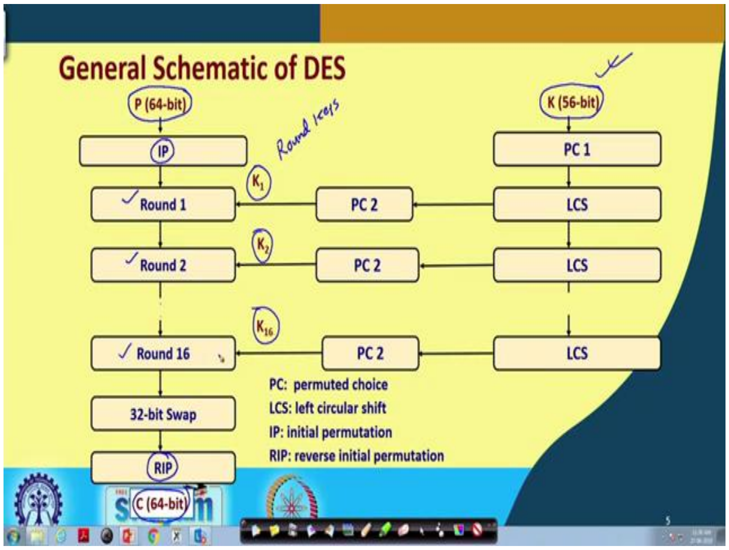


Data Encryption Standard (DES)

- The most widely used encryption scheme at one time.
  - Also known as the Data Encryption Algorithm (DEA).
  - It is a block cipher.
- Some of the features:
  - The plaintext is 64-bits in length.
  - The key is 56-bits in length.
  - Longer plaintexts are processed in 64-bit blocks.

So, let us briefly look at the data encryption standard first. So, as it said at one point in time this was most widely used. This is also sometimes referred to as Data Encryption Algorithm or DEA. As I said it is a block cipher which handles block size of 64 bits and the key size is 56 bits. So, if you have a longer plaintext, then it will be split into blocks of 64, but in general the last block can be less than 64. So, there will be some additional bits added to it, called pad to make this also 64 and accordingly using this algorithm the ciphertext will be generated, right. This is how it works.

(Refer Slide Time: 04:23)



General Schematic of DES

P (64-bit)

IP

Round 1

Round 2

Round 16

32-bit Swap

RIP

C (64-bit)

K (56-bit)

PC 1

LCS

LCS

LCS

PC 2

PC 2

PC 2

K_1

K_2

K_{16}

Round keys

PC: permuted choice
LCS: left circular shift
IP: initial permutation
RIP: reverse initial permutation

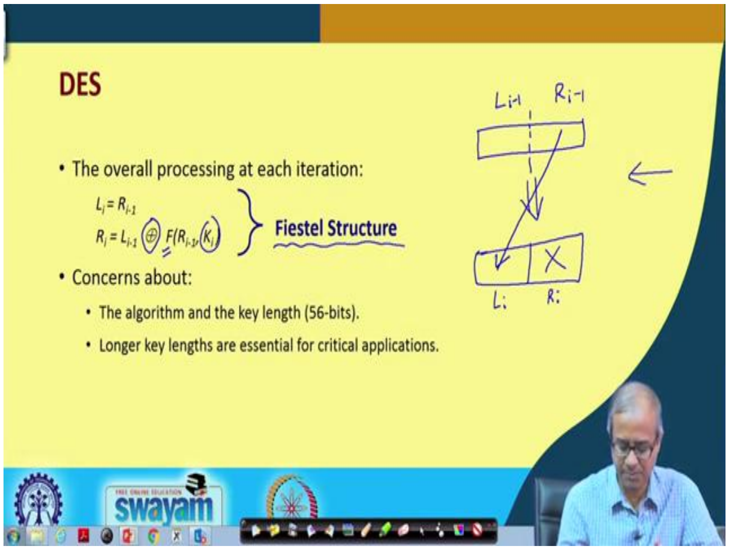
Now, I am not going to the detail explanation of how these are designed. So, I am showing you the overall schematic how this DES is internally and what are the kind of steps that are being carried out to encrypt a given plaintext. So, as I said, this is a block cipher with 64 bit of plain text so that 64 bit number is coming here at the top. And, on the other side I have the 56 bit key that is coming here. Now, you see in DES there are, there is a concept of a round. There are 16 rounds. In 16 rounds you are carrying out similar kind of calculations, but 16 times, one after the other.

So, what is the kind of calculation? First these 64 bits that is coming, you do an initial transposition, jumble up the order of the bits. This is called initial permutation. Then this initially permuted value will go through the 16 rounds. Then you do a 32 bit swap, because you have 64 bit number. You take two 32 bit chunks and interchange their order. The last you bring it to first. The first to bring it to last and then again whatever permutation you used here, you use a reverse input permutation here, the reverse permutation here and whatever comes out that will be your 64 bit ciphertext.

But another thing you see each of these rounds is dependent on another factor which is coming from the right side. These are the so called round keys. The value of the round keys are different, not the same. Although the value of K is same, PC is a permuted. Choice some kind of key permutation is going on and first permutation then LCS is left circular, circular shift, this key is undergoing a circular shift left and using some combination function you are generating the round keys.

So, round keys are changing with the round number. So, you see it is a quite complicated process through which I am generating the ciphertext from the plaintext. Now each of these rounds I am not going into the exact functionality, but broadly speaking what happens in these rounds, are like this.

(Refer Slide Time: 07:01)



DES

- The overall processing at each iteration:

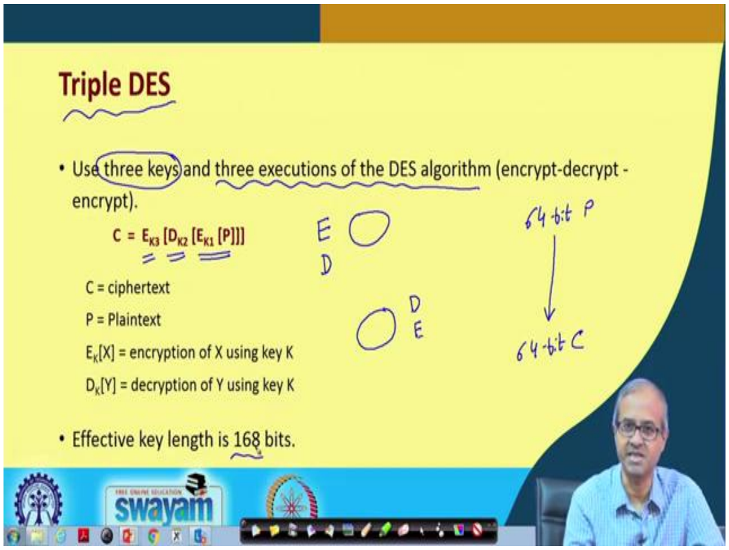
$$\left.\begin{array}{l} L_i = R_{i-1} \\ R_i = L_{i-1} \oplus F(R_{i-1}, K_i) \end{array}\right\} \text{Fiestel Structure}$$

- Concerns about:
  - The algorithm and the key length (56-bits).
  - Longer key lengths are essential for critical applications.

You see, the 64 bit plaintext that is coming and is going through each of the rounds, if you divide it to be into 2 parts, you call it a left part and a right part. So, at the ith iteration, the left part is copied with ith part. So, in the after around let us call it L_{i-1} and R_{i-1}. So, when you generate the output of this round, this will be your L_i and this will be your R_i. So, whatever this was here, this will get copied to L_i, $L_i = R_i$, but in this R_i you have a function.

This left part bit by bit exclusive-OR, some non-linear function. This is a complex function of not only R_{i-1}, but also the round key that is coming from the right side. So, this R_i is computed in a complicated way and any crypto, cryptosystem which have this kind of a, you can say structure there are many cryptosystems based on this structure, this is called in general Fiestel structure, ok. Now, as I said, DES as such is good, but the only concern is the short key length of 56 bits. For critical applications we need keys of larger sizes.

(Refer Slide Time: 08:47)



Triple DES

- Use three keys and three executions of the DES algorithm (encrypt-decrypt - encrypt).

$$C = E_{K3}[D_{K2}[E_{K1}[P]]]$$

C = ciphertext

P = Plaintext

$E_K[X]$ = encryption of X using key K

$D_K[Y]$ = decryption of Y using key K

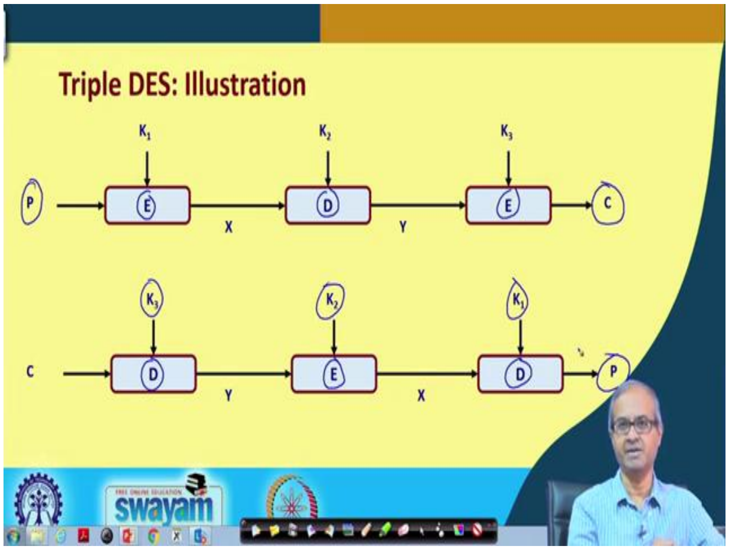
- Effective key length is 168 bits.

So, the first attempt which was done or you may say explored is to use multiple runs of the DES algorithm to provide greater security. So, here you have an algorithm which is referred to as triple DES. As the name implies triple means, you are using DES three times for every encryption. So, what do you do? There will be three different keys, 56 bit keys for DES and there will be three executions of the DES algorithm for encrypting every 64 bit plaintext. For every 64 bit plaintext, to encrypt it into a 64 bit ciphertext you will be requiring three executions or three runs of the DES algorithm. The drawback obviously, is your method will become three times slower.

But the advantage is that you are using three 56 bit keys, 56 multiplied by 3 that is essentially your strength of the key, you can say the way it works is that from the plaintext you apply the first key carry out the DES encryption process. Well decryption I have not shown. Decryption process is also quite similar, but the rounds are executed in the reverse order. So, whatever comes, you run the decryption algorithm with key K2 and whatever comes you again run the encryption algorithm with key K3.

The point to note on, note is that any data if we apply the encryption algorithm followed by the decryption algorithm you will get back the same thing. Similarly, if you apply the decryption first followed by encryption then also you will be getting back the same thing. So, this principle has been used in structuring this order of execution and as I said, 56 multiplied by three is 168. So, effective key size is 168.

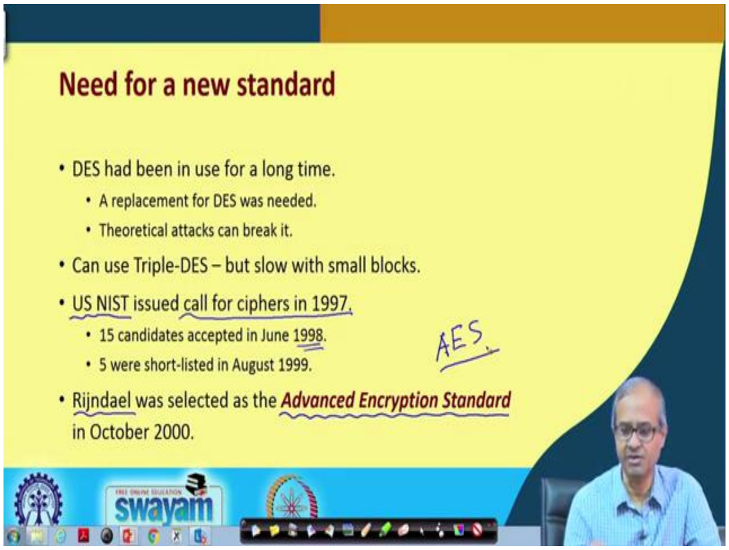
(Refer Slide Time: 11:07)


Triple DES: Illustration

Here pictorially it works like this. For encryption the plaintext is coming first. You do an encryption with the first key K_1, then decryption with K_2, then encryption with K_3. You get the final ciphertext. For decryption you do it in the reverse order. In the last step you did an encryption with K_3, you will do a decryption with K_3. In the middle you did a decryption with K_2, you do an encryption with K_2. Here you did an encryption with K_1, you do a decryption with K_1.

So, in sequence in the reverse sequence, encryption, decryption will cancel out, cancel out, cancel out and finally, you will be getting back the plaintext. This is how triple DES works. Triple DES has been also quite widely used. In fact, in many of the money ATM machines, triple DES was used till some time back. So, this was quite widely used.

(Refer Slide Time: 12:17)



Need for a new standard

- DES had been in use for a long time.
  - A replacement for DES was needed.
  - Theoretical attacks can break it.
- Can use Triple-DES – but slow with small blocks.
- US NIST issued call for ciphers in 1997.
  - 15 candidates accepted in June 1998.
  - 5 were short-listed in August 1999.
- Rijndael was selected as the *Advanced Encryption Standard* in October 2000.

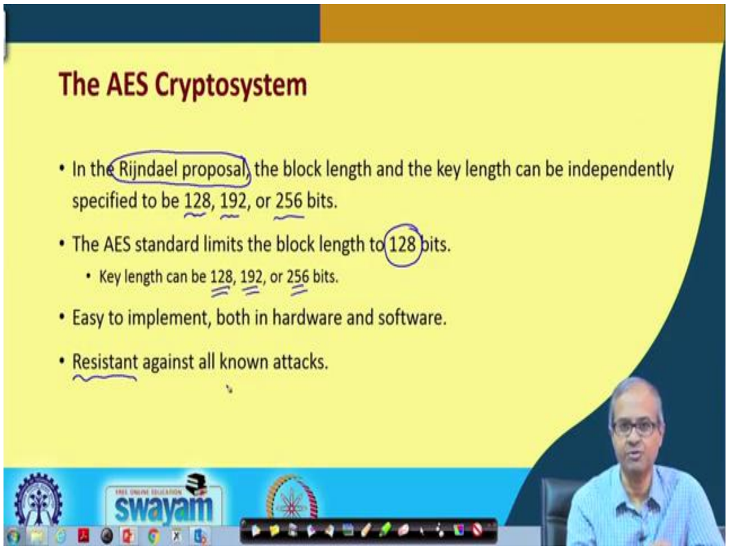
AES.

Now, there was a need for a new standard which was understood and identified, because DES was used for a pretty long time. There was no apparent weakness in the algorithm. The only concern was the key size, was smaller and many theoretical attacks designed by mathematicians were published which reduces the complexity of breaking DES. So, DES was no longer considered to be very secure. Well of course, triple DES is an option, but three executions of algorithm for every single encryption, is too much of an overhead, ok. So, that also is some kind of a drawback.

Now, the standard organization NIST in US who actually standardizes the cryptographic algorithms, they came up with a call for ciphers in 1997. They ask the mathematicians and scientists to publish their algorithms. There will be like a competition. They will be selecting the best of them as the next generation standard for encryption. So, in that way there were 15 candidates which are finally accepted in 1998, were shortlisted and finally, the Rijndael cryptosystem was selected as the advanced encryption standard, in short AES.

Sometimes we refer to this algorithm as just the AES algorithm, ok. Today this is one of the most widely used symmetric key algorithms that has been employed or deployed in many applications.

(Refer Slide Time: 14:11)



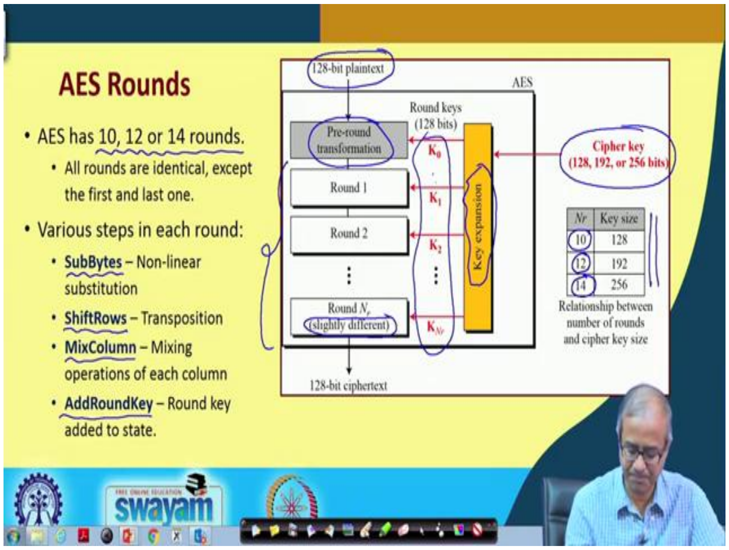
The AES Cryptosystem

- In the Rijndael proposal, the block length and the key length can be independently specified to be 128, 192, or 256 bits.
- The AES standard limits the block length to 128 bits.
  - Key length can be 128, 192, or 256 bits.
- Easy to implement, both in hardware and software.
- Resistant against all known attacks.

So, some features of this AES cryptosystem. Now, this I have already mentioned earlier, block length and the key length. Well in the, in the original Rijndael proposal which was submitted, both the block length and the key length can be independently chosen to be either 128, 192 or 256. But when NIST accepted the proposal as a standard, they added some additional constraint. They said that well let the block length be only 128, let us not have flexibility in the block length, but let the key can be either 128, 192 or 256.

So, although the original Rijndael standard can have variable block sizes, but AES will have only 128 bit block sizes. The advantage of this method is that they are very easy to implement both in hardware and also in software and there has been lot of attacks which have been attempted theoretical and hardware based attacks, but still today this algorithm is known to be resistant against all known attacks. So, it is a good and safe algorithm to use, you can say that.

(Refer Slide Time: 15:41)



AES Rounds

- AES has 10, 12 or 14 rounds.
  - All rounds are identical, except the first and last one.
- Various steps in each round:
  - SubBytes – Non-linear substitution
  - ShiftRows – Transposition
  - MixColumn – Mixing operations of each column
  - AddRoundKey – Round key added to state.

Nr	Key size
10	128
12	192
14	256

Relationship between number of rounds and cipher key size

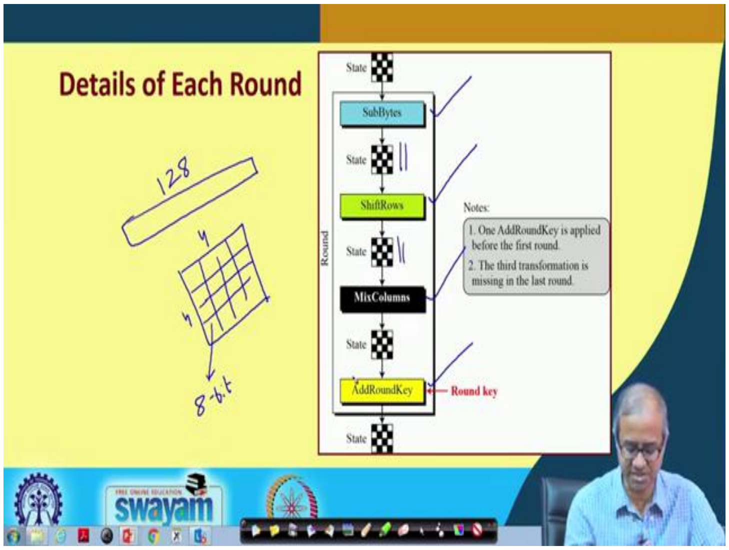
So, very briefly let us try to understand how AES works. The first thing is that depending on the key size 128, 256 or 192 the number of rounds similar to DES can vary. Now this table here shows that if key is 128 you need 10 rounds, 192 you need 12 rounds, 256 you need 14 rounds. This table shows you that. The rounds are identical except the first and last one where there is a small change that will show and each of the rounds there are some basic operations which are carried out. What are the basic operations? First operation is a substitution operation which is referred to as SubBytes.

Here every byte, you see block size is 128. So there will be 16 bytes. So, every byte is substituted by some other byte. There is substitute. Something called substitution box which is there inside. Using the SubBytes function, it replaces a byte by another byte. It is basically a permutation function. Then there is something called shift rows which is a transposition. The order is made different. the bits are not changed, but order is made different of the rows.

Then there is a mixed column operation where different columns using some operator, you combine the columns, mix them up in some certain way. So, this, there is a complicated functionality here and finally, add round key. Similar to DES, round keys are also generated here and whatever that 128 bit value is coming, you do a bit by bit exclusive-OR with this round key to generate the 128 bit data for the next round.

So, this diagram if you see, it gives the overall schematic. You start with the 128 bit plaintext which is coming. These are the rounds, but before starting with the first round you do a pre round transformation. There is an initial transformation, after that there are the rounds. The last round is slightly different. So, you see the first round is different in the sense that there is something done just before that and similarly the last round is also slightly different. And there is a key expansion module in the other side which takes as input the key 128, 192 or 256 bits and it generates the round keys similar to DES. This is how the structure of AES looks like.

(Refer Slide Time: 18:53)



Details of Each Round

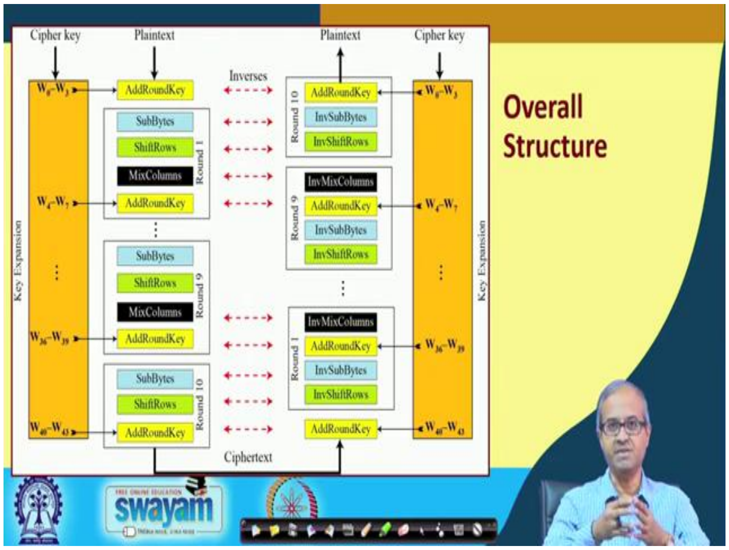
Notes:

1. One AddRoundKey is applied before the first round.
2. The third transformation is missing in the last round.

Now, if you look into each of the rounds accepting the first and the last. So, the rounds, these 4 functionalities occur in this order, substitution bytes, followed by shift rows, followed by mixed column and finally, add round key where the round key is bit by bit exclusive-OR with this number. And, another thing the 128 bit data that we are having that you are encrypting and you are moving at each stage, this is actually represented as a state vector.

There is a 4 by 4 matrix, there are 16 such states and each of them is an 8 bit quantity. 16 into 8 is 128. So, that is how this state is maintained and all these operators are defined on that state matrix. That is how it is defined in AES.

(Refer Slide Time: 20:03)


Overall Structure

And looking at the overall picture; so, here you see that in all the rounds, these operators as I said are applied in subbyte, shiftrow, mix column and add round. In the first round there is an additional add round key in the beginning and in the last round there is no mixed column states, only sub byte shift row and add round. This is how it is done. You see the structure as you can see, is fairly complicated, but these 4 basic functionalities I talked about subbytes, shiftrow, mix column and add round key, these are defined in such a way, they can be implemented very easily both using some hardware circuits. Also using some instruction set of a computer they are efficient to implement, ok.

This was one of the objectives of these algorithms, how fast they can be implemented, what is the maximum speed they can offer in various implementations, ok. So, with this we come to the end of this lecture, where we have talked about some of the practical private or symmetric key algorithms. Now, as I said whenever you are trying to secure a network, you find that there are some vulnerabilities. There are a number of ways you try to secure your infrastructure or your data or files, whatever, now this encryption is one of the very commonly used tool or technique to provide you with the required level of security.

You think of emails, you are sending mails. Some mails may be quite confidential, you may want such confidential mails should be automatically encrypted so that no one should be able to tap my messages, my mails. So, these encryption or decryption techniques can be integrated with other applications in a suitable way so that they can provide you with some levels of security as desired by some applications. In the next lecture we shall be starting some discussion on public key cryptosystems and what are the methods that can be used there.

Thank you.