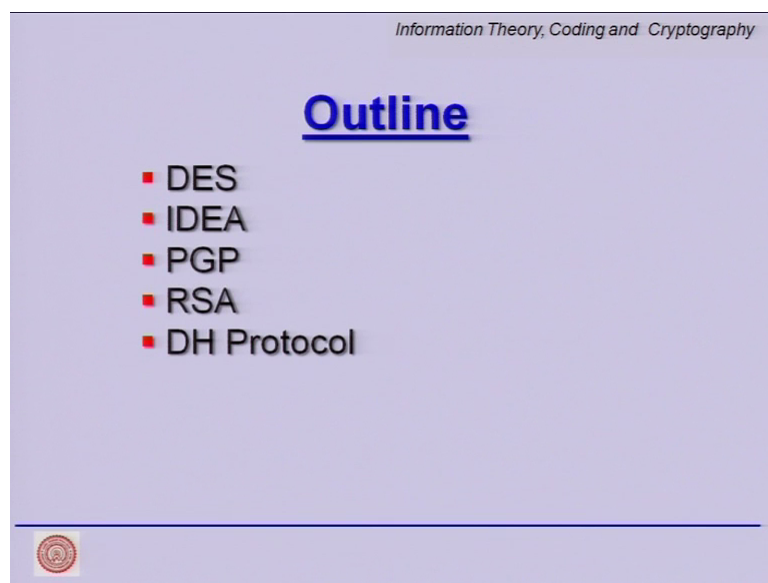


Information Theory, Coding and Cryptography
Dr. Ranjan Bose
Department of Electrical Engineering
Indian Institute of Technology, Delhi

Module - 37
Cryptography
Lecture - 37

Hello and welcome to our next lecture on Cryptography.

(Refer Slide Time: 00:32)




Let us start with the brief outline for today's talk. We would start with DES; it is an acronym for data encryption standards, then will look at couple of more acronyms idea PGP, RSA, and DH protocols. So, basically today we are going to go through a set of algorithms and protocols.

(Refer Slide Time: 01:01)

Information Theory, Coding and Cryptography

Recap

- Introduction to Cryptography
- Symmetric Key
- Asymmetric Key
- Introduction to Cryptanalysis

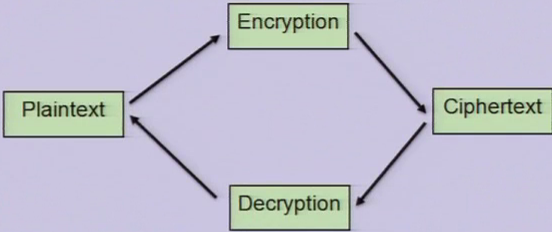


So, let us start with a quick recap what we have studied so far. We have looked at introduction to cryptography, followed by symmetric key and asymmetric key cryptography, and where a very brief introduction to cryptanalysis.

(Refer Slide Time: 01:17)

Information Theory, Coding and Cryptography

Basic Premise



```
graph LR; Plaintext -- Encryption --> Ciphertext; Ciphertext -- Decryption --> Plaintext;
```

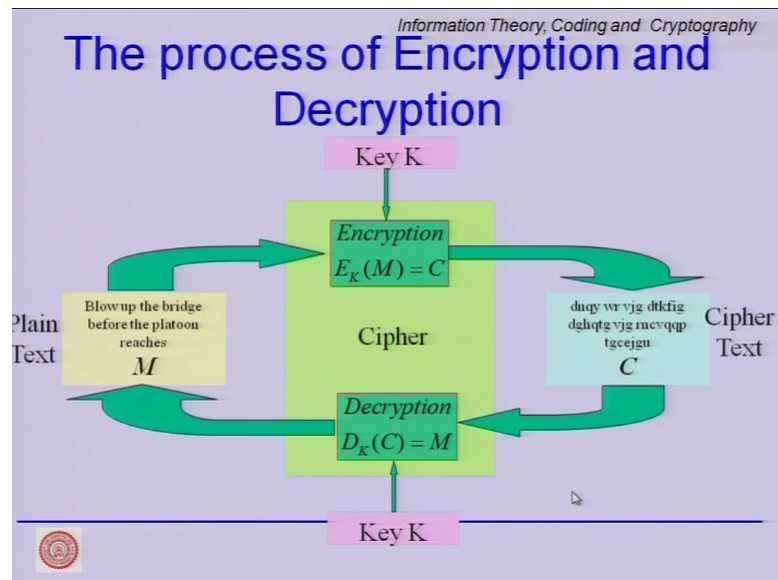
Indian Institute of Technology, Delhi

4

Ranjan Bose
Department of Electrical Engineering

Now, the basic premise is that we start with the plaintext go for encrypt, and obtain a ciphertext which is not discernable to the unintended users. But that intended valid user goes to decryption process and gets the plaintext back.

(Refer Slide Time: 01:40)



We looked at the symmetric key encryption, we start with the plaintext again we go through the encryption block using one key, and then we receive the ciphertext the ciphertext undergoes a decryption using the same key and we get the plaintext. So, this was the symmetric key or the secret key encryption method, because the key needs to be kept secret in order to preserve the authenticity of the message.

(Refer Slide Time: 02:08)

Information Theory, Coding and Cryptography

Types of symmetric algorithms

- There are two types of **symmetric** algorithms, block ciphers and stream ciphers.
 - **Block ciphers** usually operate on groups of bits called blocks. Each block is processed a multiple number of times. In each round the key is applied in a unique manner. The more the number of iterations, the longer is the encryption process, but results in a more secure ciphertext.
 - **Stream ciphers** operate on plaintext one bit at a time. Plaintext is streamed as raw bits through the encryption algorithm. While a block cipher will produce the same ciphertext from the same plaintext using the same key, a stream cipher will not. The ciphertext produced by a stream cipher will vary under the same conditions.

A small circular logo is visible in the bottom left corner of the slide.

Now, we looked at symmetric key algorithms, which are of 2 types block ciphers where we process block by block and encrypt each block, using a key or a set of keys, or sub


keys whereas stream cipher works. It is like the convolution encoder we have seen it is with memory and that the current output depends on the past inputs and current input.

(Refer Slide Time: 02:39)

Information Theory, Coding and Cryptography

Data Encryption Standard (DES)

- DES, an acronym for the Data Encryption Standard, is the name of the Federal Information Processing Standard (FIPS) 46-3, which describes the data encryption algorithm (DEA).
- The DEA is also defined in the ANSI standard X9.32.
- Created by IBM, DES came about due to a public request by the US National Bureau of Standards (NSB) requesting proposals for a standard cryptographic algorithm that satisfied the following criteria:

 Indian Institute of Technology,
Delhi

7

Ranjan Bose
Department of Electrical Engineering


So now we go on to look at some of the standards past present in future which help us understand how we use the theory in practice. So, we looked at this data encryption standard from the classical perspective, because this is kind of now obsolete it has been replaced by AES advanced encryption standard, but for classical reasons let us look at it. So, basically it was created by IBM in the 70's.

(Refer Slide Time: 03:10)

Information Theory, Coding and Cryptography

DES

- Provides a high level of security
- The security depends on keys, not the secrecy of the algorithm
- The security is capable of being evaluated
- The algorithm is completely specified and easy to understand
- It is efficient to use and adaptable
- Classically important:
- **Advanced Encryption Standard (AES) is now used worldwide. It supersedes the Data Encryption Standard (DES)**

 Indian Institute of Technology,
Delhi

8

Ranjan Bose
Department of Electrical Engineering

And what it did provide and it still does provide is a very high level of security with limited key size.

So, remember the security depends on the key and not on the secrecy of the algorithm so, D is algorithm is out in the open, right. So, this is a very efficient algorithm classically important and it is most importantly the stepping stone for the next generation encryption standards. So, the next generation is the advanced encryption standard AES which is now use worldwide and it is supersedes the DES or the data encryption standard with that background.

(Refer Slide Time: 03:51)

Information Theory, Coding and Cryptography

DES

- DES is a **symmetric, block-cipher algorithm** with a key length of 64 bits, and a block size of 64 bits (i.e. the algorithm operates on successive 64 bit blocks of plaintext).
- Being **symmetric**, the same key is used for encryption and decryption, and DES also uses the same algorithm for encryption and decryption.
- **So is AES !**

Indian Institute of Technology, Delhi

9

Ranjan Bose
Department of Electrical Engineering

Let us quickly look at what DES does and how does it work. So, it is a symmetric block cipher algorithm, we have studied what are block cipher algorithms.


So, the key length typically is 64 bits, and it works with a block size of 64 bits. So, we take a non-coded plaintext of 64-bit block, and then work with it using a 64 bit. Being symmetric the same key that is used for encryption, and is used for description and same as the setting for AES, but today we will just discuss DES in detail.

(Refer Slide Time: 04:32)

Information Theory, Coding and Cryptography

DES

- First a **transposition** is carried out according to a set table (the initial permutation), the 64-bit plaintext block is then split into two 32-bit blocks, and 16 identical operations called rounds are carried out on each half.
- The two halves are then **joined back together**, and the reverse of the initial permutation carried out.

 Indian Institute of Technology, Delhi 10 Ranjan Bose
Department of Electrical Engineering


So, what are the steps? The first step is a transposition is carried out according to the set up a table, which is the initial permutation. So, the 64-bit plaintext block is; first, split into 2 32 bit blocks. And then 16 identical operations call rounds are carried out on each halves, ok. This is made public people know how to do this, right. And then the 2 halves of 32 bit block a joined back together.

(Refer Slide Time: 05:05)

Information Theory, Coding and Cryptography

DES

- The **64-bit key is reduced to 56** by removing every eighth bit (these are sometimes used for error checking).
- Sixteen different 48-bit subkeys are then created - one for each round.
- This is achieved by splitting the 56-bit key into two halves, and then circularly shifting them left by 1 or 2 bits, depending on the round.
- After this, 48 of the bits are selected.
- Because they are shifted, different groups of key bits are used in each subkey.
- This process is called a **compression permutation** due to the transposition of the bits and the reduction of the overall size.

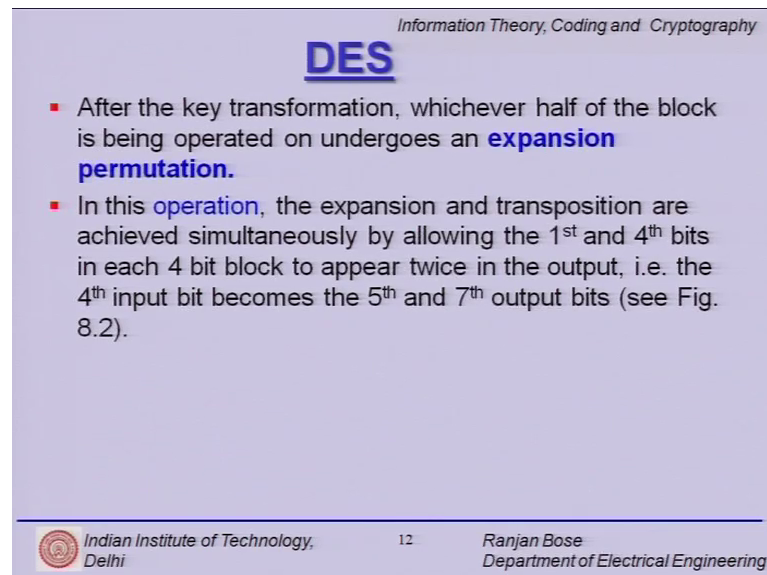
 Indian Institute of Technology, Delhi 11 Ranjan Bose
Department of Electrical Engineering

The 64 bit keys first reduced to 56, bits' key by removing every 8 bit, and these are probably later used for error checking. So, 16 different 48-bit sub keys are then created,

one for each round so, we talked about those earlier 16 rounds. So, this is achieved by splitting the 56-bit key into 2 halves, and then circularly shifting them left by 1 or 2 bits. So, these are the details basically what it tells us is that we sub divided the block, and then we take the keys, and then we do the permutations and then finally, we carry out the encoding.

So, different groups of key bits are used as different sub keys. The process that we just not discussed is called compression permutation, due to the transposition of the bits and the reduction of the overall size, we call it as a compression permutation.

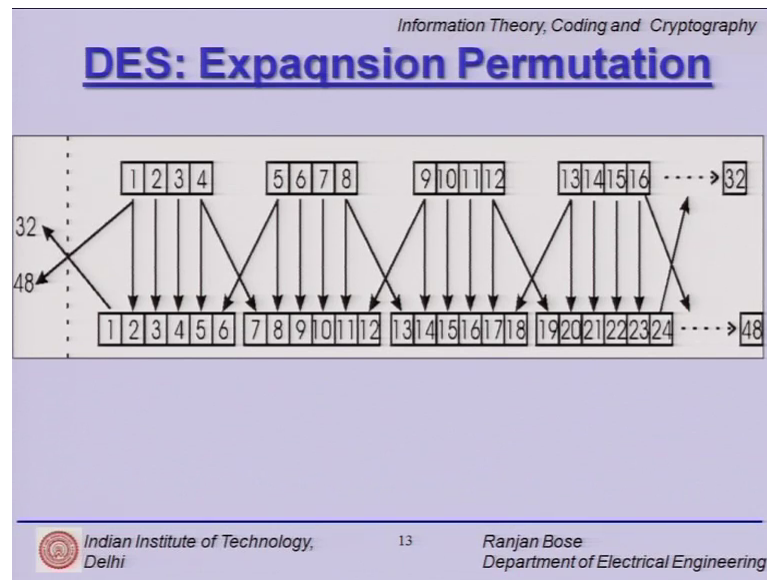
(Refer Slide Time: 06:08)



The slide is titled "DES" in a large, bold, blue font. Above the title, it says "Information Theory, Coding and Cryptography". Below the title, there are two bullet points. The first bullet point says: "After the key transformation, whichever half of the block is being operated on undergoes an **expansion permutation**." The second bullet point says: "In this **operation**, the expansion and transposition are achieved simultaneously by allowing the 1st and 4th bits in each 4 bit block to appear twice in the output, i.e. the 4th input bit becomes the 5th and 7th output bits (see Fig. 8.2)." At the bottom of the slide, there is a footer with the Indian Institute of Technology Delhi logo, the text "Indian Institute of Technology, Delhi", the number "12", and the name "Ranjan Bose, Department of Electrical Engineering".

After the key transformation, whichever half blocked is being operated upon undergoes an expansion permutation; now, after the compression operation. So, details are as follows is how you can basically in this operation the expansion transpositions are achieved simultaneously by allowing forth and the first bit in each block to appear twice in the output. Therefore, we replicate and therefore we expand.

(Refer Slide Time: 06:35)



We can look at it in terms of a diagram. So, you have these 4 bits, and this fourth bit is replicated some. So, with fifth bit is replicate so, we kind of expand. So, there is an expansion permutation this is a spelling mistake.

(Refer Slide Time: 06:52)

Information Theory, Coding and Cryptography

DES

- **The expansion permutation achieves 3 things:** Firstly it increases the size of the half-block from 32 bits to 48, the same no of bits as in the compressed key subset, which is important as the next operation is to XOR the two together.
- Secondly, it produces a **longer string of data** for the substitution operation that subsequently compresses it.
- Thirdly, and most importantly, because in the subsequent substitutions the 1st and 4th bits appear in two S-boxes (described shortly), they affect two substitutions.
- The effect of this is that the dependency of the output bits on the input bits **increases rapidly**, and so therefore does the security of the algorithm.

Indian Institute of Technology, Delhi 14 Ranjan Bose
Department of Electrical Engineering

Here, but what does this (Refer Time: 06:55) this expansion permutation achieves 3 things. Firstly, it increases the size from 32 bits to 48 bits.

It produces a longest string of data, for the substitution operation that subsequently compresses it, and most importantly because in the subsequent substitutions, the first and

the forth bits appearing to S boxes; which will describe shortly S boxes stand for substitution boxes. And the effect of is this is that the dependency of the output bits on the input bits increases rapidly. And so, therefore, does the security of the algorithm.

(Refer Slide Time: 07:36)

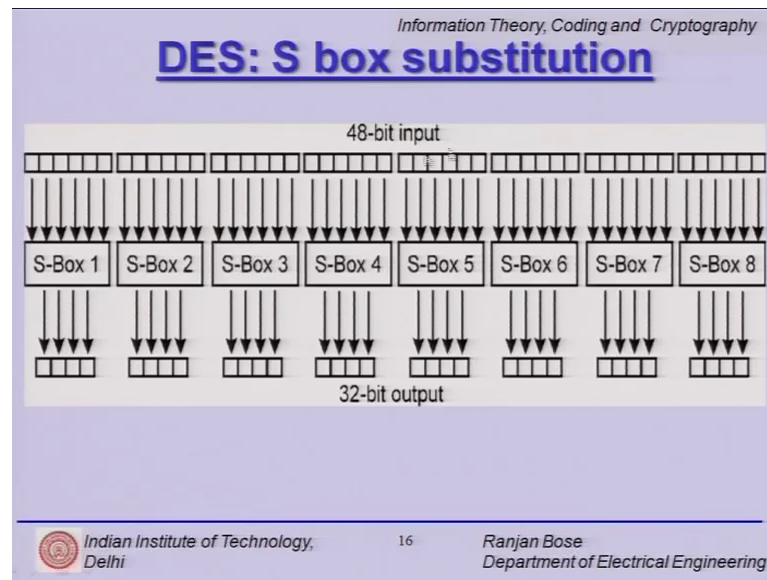
The slide is titled "DES" in a large, bold, blue font. Below the title, the text "Information Theory, Coding and Cryptography" is written in a smaller, grey font. The main content consists of five bullet points, each starting with a red square. The first bullet point states: "The next operation is to perform **substitutions on the expanded block**." The second bullet point states: "There are eight substitution boxes, called **S-boxes**". The third bullet point states: "The first S-box operates on the first 6 bits of the 48-bit expanded block, the 2nd S-box on the next six, and so on." The fourth bullet point states: "Each S-box operates from a table of 4 rows and 16 columns, each entry in the table is a 4-bit number." The fifth bullet point states: "The net result of the **substitution phase** is eight 4-bit blocks that are then combined into a 32-bit block." At the bottom of the slide, there is a footer containing the Indian Institute of Technology Delhi logo, the text "Indian Institute of Technology, Delhi", the page number "15", and the name "Ranjan Bose, Department of Electrical Engineering".

- The next operation is to perform **substitutions on the expanded block**.
- There are eight substitution boxes, called **S-boxes**
- The first S-box operates on the first 6 bits of the 48-bit expanded block, the 2nd S-box on the next six, and so on.
- Each S-box operates from a table of 4 rows and 16 columns, each entry in the table is a 4-bit number.
- The net result of the **substitution phase** is eight 4-bit blocks that are then combined into a 32-bit block.

So, after this in DES, the next operation is substitutions on the expanded blocks, there are 8 substitution boxes called S boxes, the first S box operates on the first 6 bits of the 48-bit expanded block, similarly the second S box on the next 6 and so on and so forth. Again these are the details these are openly available. And the net result of the substitution phase is, that 8 4 bit blocks that are then combined into 30, 4, 2-bit block.

So, basically we break it up do the operations do the permutations, do the expansion then combine them back into 32-bit block. This is these are the details of the operations.

(Refer Slide Time: 08:20)



So, this is how this 8 S boxes work with you start with 48 bits, and divided into S boxes and then go on to get the 32-bit output.

(Refer Slide Time: 08:32)

Information Theory, Coding and Cryptography

DES

- The 32-bit output of the substitution phase then undergoes a **straightforward transposition** using a table sometimes known as the P-box.
- After all the rounds have been completed, the two 'half-blocks' of 32 bits are recombined to form a 64-bit output, the final permutation is performed on it, and the resulting 64-bit block is the **DES encrypted ciphertext** of the input plaintext block.

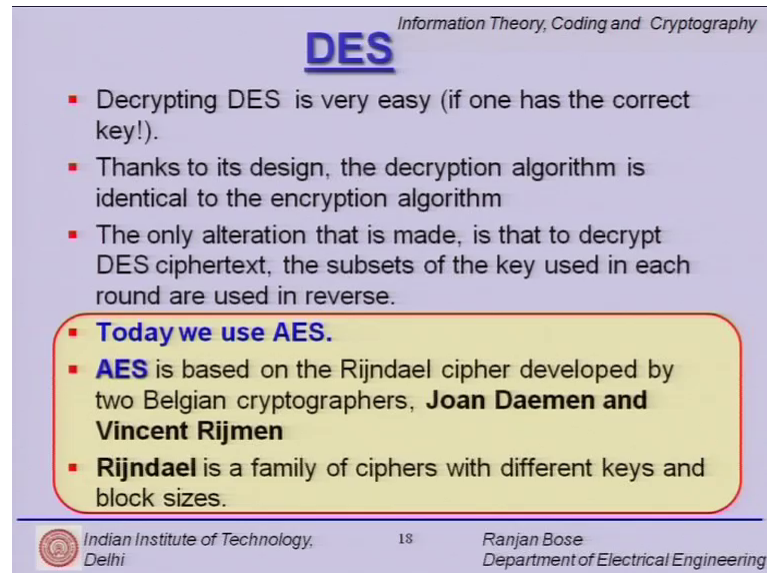
Indian Institute of Technology, Delhi 17 Ranjan Bose Department of Electrical Engineering

So, the 32-bit output of the substitution phase that undergoes a straightforward transposition, and using a table sometimes known as the P box for permutation box. And basically after all the rounds that have been completed the 2 half box remember we started off with 2 32-bit half are recombined to form a 64-bit output. The final

permutations performed on it and the resulting 64-bit block is finally, the DES encrypted ciphertext, ok.

So, just a series of steps can be encoded into a computer program very easily.

(Refer Slide Time: 09:11)




Information Theory, Coding and Cryptography

DES

- Decrypting DES is very easy (if one has the correct key!).
- Thanks to its design, the decryption algorithm is identical to the encryption algorithm
- The only alteration that is made, is that to decrypt DES ciphertext, the subsets of the key used in each round are used in reverse.

- **Today we use AES.**
- **AES** is based on the Rijndael cipher developed by two Belgian cryptographers, **Joan Daemen and Vincent Rijmen**
- **Rijndael** is a family of ciphers with different keys and block sizes.

 Indian Institute of Technology,
Delhi18Ranjan Bose
Department of Electrical Engineering

So, decryption part is quite easy, provided you have the right key, thanks to its design the decryption algorithm is identical, but in the reverse order of the encryption algorithm. So, whatever we do splitting we do combining we do substitution, we can undo that in the decryption process. The only alteration that is made is to decrypt DES ciphertext, the subsets of the key used in each rounds are used in the reverse order. DES is really no longer used; today we use AES, but you have understood the concept AES is based on rijndael cipher developed by 2 Belgian cryptographer Joan Daemen and Vincent Rijmen. And this rijndael cipher that we use in AES is belongs to a family of ciphers with different keys and block sizes.


This in general gives a bird's eye view so, to say of the DES how it does and how you can go and make the different steps in order.

(Refer Slide Time: 10:19)

Information Theory, Coding and Cryptography

International Data Encryption Algorithm (IDEA)

- IDEA was created in its first form by Xuejia Lai and James Massey in 1990, and was called the **Proposed Encryption Standard (PES)**.
- In 1991, Lai and Massey strengthened the algorithm against differential cryptanalysis and called the result **Improved PES (IPES)**.
- The name of IPES was changed to International Data Encryption Algorithm (**IDEA**) in 1992.
- IDEA is perhaps best known for its implementation in **PGP (Pretty Good Privacy)**.
- The last patents expired in 2012 and IDEA is now **patent-free and thus free to use**

 Indian Institute of Technology, Delhi

19

Ranjan Bose
Department of Electrical Engineering

We now move on to the next so, to say defect to standard, which is called international data encryption algorithm, or the acronym is idea what does it do? Well, it was created in 1990's, and it was originally called the proposed encryption standard even after it was proposed. And then in the sub following year it was strengthened and it was called the improved proposed encryption standard or IPES.


Later on the next year the name was formerly change to international data encryptions algorithm idea, ok. And that is how it is known. And we know idea as being used in PGP or the pretty good privacy which is pretty common for securing our emails, we will spend one slide on PGP later on in this talk. Well, the good news is in 2012, the patents regarding idea expired and now, it is patent free and thus free to use. So, we can expect a lot of proliferation for this idea.

(Refer Slide Time: 11:30)

Information Theory, Coding and Cryptography

IDEA

- The algorithm IDEA is a symmetric, block-cipher algorithm with a key length of 128 bits, a block size of 64 bits, and as with DES, the same algorithm provides encryption and decryption.
- IDEA consists of 8 rounds using 52 subkeys.
- Each round uses six subkeys, with the remaining four being used for the output transformation.

 Indian Institute of Technology, Delhi 20 Ranjan Bose
Department of Electrical Engineering


So, let us quickly spend a few minutes on looking at overall how idea works. First of all, idea is a symmetric block cipher, with a key length of 128 bits. So, immediately the strength of the algorithm goes up because the key length is pretty good. So, block size use a 64 bits, and as with DES the algorithm provides the encryption and decryption. So, it is a symmetric idea again has some rounds actually 8 rounds using 52 sub keys rest of the matter of the detail how these rounds are executed and how these sub keys are found, each round use a 6 sub keys with the remaining 4 being used for the output.

(Refer Slide Time: 12:14)

Information Theory, Coding and Cryptography

IDEA

- Firstly the 128-bit key is divided into eight 16-bit keys to provide the first eight subkeys.
- The bits of the original key are then shifted 25 bits to the left, and then it is again split into eight subkeys.
- This shifting and then splitting is repeated until all 52 subkeys (SK1-SK52) have been created.

 Indian Institute of Technology, Delhi 21 Ranjan Bose
Department of Electrical Engineering


So firstly, what do we do with this 128-bit key, is divided into 8 16 bit keys to provide the first 8 sub keys. The bits of the original key are then shifted 25 bits to the left, and then it is again split into 8 sub keys, these details can be worked out, and the shifting and then splitting is repeated until all 52 sub keys have been created.

(Refer Slide Time: 12:40)

Information Theory, Coding and Cryptography

IDEA

- The 64-bit plaintext block is firstly split into **four blocks** (B1-B4).
- A round then consists of the following steps (**OB stands for output block**):
 - $OB1 = B1 * SK1$ (multiply 1st sub-block with 1st subkey)
 - $OB2 = B2 + SK2$ (add 2nd sub-block to 2nd subkey)
 - $OB3 = B3 + SK3$
 - $OB4 = B4 * SK4$ (multiply 3rd sub-block with 3rd subkey)
 - $OB5 = OB1 \text{ XOR } OB3$ (XOR results of steps 1 and 3)
 - $OB6 = OB2 \text{ XOR } OB4$

 *Indian Institute of Technology, Delhi* 22 *Ranjan Bose
Department of Electrical Engineering*

So, what do we do with them? The 64-bit plaintext block that we have to encode is first split into 4 blocks. So, remember the key length was 128, which was used to generate sub keys, but the block length for the un-coded the plaintext is 64 bits, which first divided into 4 16 bit blocks B 1 to B 4.

And then we have to do this keys, somehow operate the keys on the blocks to get output block. So, for example, the output block one takes the first sub block of the plaintext and multiplied exhort, rather with the first key, similarly second one is the second key; so we of these operations that you can carry out up to this 14 output blocks.


(Refer Slide Time: 13:30)

Information Theory, Coding and Cryptography

IDEA

- $OB7 = OB5 * SK5$ (multiply result of step 5 with 5th subkey)
- $OB8 = OB6 + OB7$ (add results of steps 5 and 7)
- $OB9 = OB8 * SK6$ (multiply result of step 8 with 6th subkey)
- $OB10 = OB7 + OB9$
- $OB11 = OB1 \text{ XOR } OB9$ (XOR results of steps 1 and 9)
- $OB12 = OB3 \text{ XOR } OB9$
- $OB13 = OB2 \text{ XOR } OB10$
- $OB14 = OB4 \text{ XOR } OB10$

23

 Indian Institute of Technology,
DelhiRanjan Bose
Department of Electrical Engineering

Again very, very easy to implement in hardware, ok, these are all multiplications.


(Refer Slide Time: 13:40)

Information Theory, Coding and Cryptography

IDEA

- The input to the next round, is the **four sub-blocks** OB11, OB13, OB12, OB14 in that order.
- After the eighth round, the four final output blocks (F1-F4) are used in a final transformation to produce four sub-blocks of ciphertext (C1-C4) that are then rejoined to form the final 64-bit block of ciphertext.
- $C1 = F1 * SK49$
- $C2 = F2 + SK50$
- $C3 = F3 + SK51$
- $C4 = F4 * SK52$
- Ciphertext = C1 & C2 & C3 & C4.

24

 Indian Institute of Technology,
DelhiRanjan Bose
Department of Electrical Engineering

So now what do we do? In the input to the next round, we complete the first round, the 4 sub blocks are used in that order O B 1 1 O B 1 3 O B 1 2 and O B 1 4. And after the 8 round 4 final output blocks F 1 to F 4 are used for the final transformation to produce the 4 sub blocks of the ciphertext. And how are they formed? Again C 1 is F 1 star S K 49 and so on and so forth. And you get the ciphertext eventually C 1 C 2 C 3 and C 4, these are the ciphertext blocks.

(Refer Slide Time: 14:25)

Information Theory, Coding and Cryptography

IDEA - Advantages

- Not only is IDEA approximately **twice as fast** as DES, but it is also considerably more secure.
- Using a brute-force approach, there are 2^{128} possible keys.
- If a **billion** chips that could each test 1 billion keys a second were used to try and crack an IDEA-encrypted message, it would take them **10^{13} years** which is considerably longer than the age of the universe!

Indian Institute of Technology, Delhi 25 Ranjan Bose
Department of Electrical Engineering

Now, let us look at the advantages, first of all it is the speed, very good for real time operations where comparing it with DES for classical reasons, but the point remains the same DES almost twice as fast as. So, a idea is twice as fast as DES, but at the same time more secure. So, brute force approach would require you to try all 2^{128} possible keys which is huge. I mean, if you look at one billion key attempts per second it will take you 10^{13} years which is more than the age of the universe.

(Refer Slide Time: 15:06)

Information Theory, Coding and Cryptography

RC Ciphers

- The RC ciphers were designed by Ron Rivest for the RSA Data Security.
- RC stands for *Ron's Code* or *Rivest Cipher*.
- **RC2** was designed as a quick-fix replacement for DES, that is more secure.
- It is a block cipher with a variable key size that has a propriety algorithm.
- RC2 is a variable-key-length cipher.
- However, when using the Microsoft Base Cryptographic Provider, the key length is hard-coded to 40 bits.
- When using the Microsoft Enhanced Cryptographic Provider, the key length is 128 bits by default and can be in the range of 40 to 128 bits in 8-bit increments.

Indian Institute of Technology, Delhi 26 Ranjan Bose
Department of Electrical Engineering

So, it is pretty secure. Let us spent couple of more minutes on the next family of ciphers called the RC ciphers, they were designed by Ron Rivest, for the RSA data security. They also called Ron's code or Rivest cipher, that that is the RC acronym. So, there is RC1 RC2 RC3 RC4 so and so forth. RC2 was designed as a quick fix replacement for DES that is more secure more efficient. It is a block cipher with a variable key size and has a proprietary algorithm. So, innovate is the disadvantage, RC2 is a variable key length cipher. And so, it is used also in the Microsoft base cryptographic provider, the key length is 40 bits hardwired, and when using this Microsoft enhance cryptographic provider, the key length could be put to 128 bits by default, right.

(Refer Slide Time: 16:17)

The slide is titled "RC4" and is part of a presentation on "Information Theory, Coding and Cryptography". It contains a bulleted list of facts about RC4 and RC5. At the bottom, it includes the logo of the Indian Institute of Technology, Delhi, the slide number 27, and the name of the lecturer, Ranjan Bose, from the Department of Electrical Engineering.

RC4

- **RC4** was developed by Ron Rivest in 1987.
- It is a variable-key-size stream cipher.
- The details of the algorithm have not been officially published.
- The algorithm is extremely easy to describe and program.
- Just like RC2, 40-bit RC4 is supported by the Microsoft Base Cryptographic provider, and the Enhanced provider allows keys in the range of 40 to 128 bits in 8-bit increments.
- **RC5** is a block cipher designed for speed.
- The block size, key size and the number of iterations are all variables.
- In particular, the key size can be as large as 2,048 bits.

Indian Institute of Technology, Delhi 27 Ranjan Bose Department of Electrical Engineering

Similarly, we have another example of RC4, it has a variable key size, but this is now stream cipher as a post to the block cipher. So, we do not really know the details of the algorithm, but it is very easy to describe and program. And just like RC2 it is also supported by Microsoft base cryptographic provider. The next version RC5 is a block cipher again designed for speed. So, basically there is a tradeoff between speed and security, and the key length of course, the computational complexity also comes out. So, RC5 has a variable block size key size number of iterations, all of them can be fixed by the user.


What is a very interesting is that the key size, which has a direct implication on the security can be as large as 2048 bits.

(Refer Slide Time: 17:19)

Information Theory, Coding and Cryptography

Public-Key Encryption

- **Public-key Algorithms** are **asymmetric**, that is to say the key that is used to encrypt the message is different to the key used to decrypt the message.
- The encryption key, known as the **public key** is used to encrypt a message, but the message can only be decoded by the person that has the decryption key, known as the **private key**.
- This type of algorithm has a number of advantages over traditional symmetric ciphers.
- It means that the recipient can make their public key widely available - anyone wanting to send them a message uses the algorithm and the recipient's public key to do so.
- An eavesdropper may have both the algorithm and the public key, but will still not be able to decrypt the message.
- Only the recipient, with their private key can decrypt the message.

 *Indian Institute of Technology, Delhi* 28 *Ranjan Bose
Department of Electrical Engineering*

So, far we have studying these private key encryptions, symmetric key where the same key is used for encoding and decoding. We now change gears and we look at the public key encryption where we have the notion of a public key and a private key. We talked about it briefly in the previous lecture, but let us look at some examples and considerate in more detail. So, public key and algorithms by definition are asymmetric. So, the key used to encrypt is definitely different from the key used to decrypt. So, the encryption key is called the public key, where is the decryption is called the private key.

This type of algorithm has a number of advantages, right, because the key transfer key exchange it is really much more efficient in this case, ok. So, the basic idea is in the recipient can make his or her public key available widely. And anybody wants to send the message can use the public key to encode and send, whereas it can only be opened by the private key. So, only the recipient intended recipient legitimate receiver with the private key can decrypt the message.

(Refer Slide Time: 18:45)

Information Theory, Coding and Cryptography

Public-Key Encryption

- A disadvantage of public-key algorithms is that they are more **computationally intensive** than symmetric algorithms, and therefore encryption and decryption take longer.
- This may not be significant for a short text message, but certainly is for long messages or audio/video.
- The **Public-Key Cryptography Standards** (PKCS) are specifications produced by **RSA Laboratories** in cooperation with secure systems developers worldwide for the purpose of accelerating the deployment of public-key cryptography.

Indian Institute of Technology, Delhi 29 Ranjan Bose
Department of Electrical Engineering

So, disadvantage of public key algorithms is that they are more computationally intensive. So, it is short messages for key exchange, but not for large volumes of messages. So, public key standards the P K C S are specifications produced by the RSA labs in cooperation with secure system developers worldwide.

(Refer Slide Time: 19:10)

Information Theory, Coding and Cryptography

RSA Algorithm

- **RSA**, named after its three creators - **Rivest, Shamir and Adleman**, was the first effective public-key algorithm, and for years has withstood intense scrutiny by cryptanalysts all over the world.
- Unlike symmetric key algorithms, public-key algorithms rely on it being **computationally unfeasible** to recover the private key from the public key.

Indian Institute of Technology, Delhi 30 Ranjan Bose
Department of Electrical Engineering

So, let us now talk about one specific example called the RSA algorithm, named after Rivest Shamir and Adleman, the 3 inventors right. So, the first effective public key algorithms and it has stood the test of time. So, public key algorithms has we know rely

on being computationally unfeasible to recover the private key, from the public key that is the basic premise it is a computational difficulty which prevents the hacker deciphering the message.

(Refer Slide Time: 19:42)


Information Theory, Coding and Cryptography

RSA Algorithm

- RSA relies on the fact that it is **easy to multiply** two large prime numbers together, but extremely hard (i.e. time consuming) to **factor them back** from the result.
- Factoring a number means finding its prime factors, which are the prime numbers that need to be multiplied together in order to produce that number.

For example:

$$10 = 2 \times 5$$
$$60 = 2 \times 2 \times 3 \times 5$$
$$2^{113} - 1 = 3391 \times 23279 \times 65993 \times 1868569 \times 1066818132868207$$

 Indian Institute of Technology, Delhi

31

Ranjan Bose
Department of Electrical Engineering

So, what does RSA rely on? The basic idea is that it believes that it is easy to multiply 2 large prime numbers, but extremely hard to factor them back. So, the one-line motivation regarding RSA easy to multiply difficult to factor factoring a number means finding its prime factors. So, for example, here we know 10 can be written as into 5, 60 again can be broken up and these are obvious, but the moment I go to slightly larger number like 2 raise power 113 minus 1 factorization becomes extremely hard.


So, in this example, you can see going from multiplication going from the right hand side to the left hand side is very easy I can multiply them and get you this big number. But given this huge number, getting the factors is a hard task. In fact, some of the good efficient computers a tested for this speed by giving them difficult numbers large numbers to be factorized and to check whether they are prime numbers.

(Refer Slide Time: 21:01)

Information Theory, Coding and Cryptography

RSA Algorithm

- Two very large prime numbers, normally of equal length, are randomly chosen then multiplied together.
- $N = A \times B$
- $T = (A - 1) \times (B - 1)$.
- A third number is then also chosen randomly as the public key (E) such that it has no common factors (i.e. is relatively prime) with T .
- The private key (D) is then:
- $D = E^{-1} \text{ mod } T$
- To encrypt a block of plaintext (M) into ciphertext (C):
- **$C = M^E \text{ mod } N$.**
- To decrypt:
- **$M = C^D \text{ mod } N$.**

 Indian Institute of Technology,
Delhi32Ranjan Bose
Department of Electrical Engineering

So, how do we make this RSA algorithm work? The algorithm is as follows, to very large prime numbers large, how large? Are there enough last prime numbers? Which are those questions, but let us assume that there enough large prime numbers will ask those questions, but let us assume that there are enough large prime numbers.

So, we pick 2 of them to very large prime numbers normally of similar lengths are randomly chosen, and then multiplied together. So, A and B are multiplied to get N. And then we get another number T as a product of a minus 1 times B minus 1. So, third number is also chosen randomly as the public key E so, it is chosen. Such that it is no common factor, what does it mean? That is relatively prime with T. So, it has no common factors with T, and how did we get T? A minus 1 into B minus 1, what is A? One of the prime numbers picked at random B is the other prime number picked at random, A and B are of similar sizes.

So, how do we get the private key? Because the public key was chosen randomly; so to get the private key we compute D as E inverse mod T. And to encrypt a plaintext M we take M raise per E mod N. So, please note, what are we doing? M into M into M so and so forth E times; therefore, we get M raise per E, but remember we are working with large large numbers, so, E can be really really large. So, self-multiplication many, many, many number of times could be difficult. And then of course, we have modulo M, how do we decrypt? Well to decrypt getting back the message M from the ciphertext C, we

take C, and then multiply with itself D times. What is D? D is the private key, and this is also very large.


So, again you can see computationally it is really involved.

(Refer Slide Time: 23:18)

Information Theory, Coding and Cryptography

Example

- 1st prime (A) = 37
- 2nd prime (B) = 23
- So,
- $N = 37 \cdot 23 = 851$
- $T = (37 - 1) \times (23 - 1) = 36 \times 23 = 792$
- E must have no factors other than 1 in common with 792.
- E (public key) could be 5.
- D (private key) = $5^{-1} \text{ mod } 792 = 317$
- To encrypt a message (M) of the character 'G':
- If G is represented as 7 (7th letter in alphabet), then $M = 7$.
- C (ciphertext) = $7^5 \text{ mod } 851 = 638$ **$C = M^E \text{ mod } N.$**
- To decrypt:
- $M = 638^{317} \text{ mod } 851 = 7.$ **$M = C^D \text{ mod } N.$**

 *Indian Institute of Technology, Delhi* 33 *Ranjan Bose
Department of Electrical Engineering*

Let us take a very simple example; this is a toy example because the prime numbers will never be so small they will run across the slide. So, let us take prime number a as 37 prime number B as 23. So, we get N as a product of these 2 prime numbers, similar T is a minus 1, into B minus 1, and we multiply and we get this number. Now we must choose the public key E, the conditions is that the public key must have no factors other than one common with this T. So, again a very small number 5, we find out that 5 is not a factor of 7 9 2. And easily we can pick E as a valid public key 5.

So, to generate D which is the private key we take 5 inverse, mod 7 9 2 and you get this number, alright. So, once you calculate D, you can now go for getting a ciphertext. So, suppose we want to encrypt G a letter G. So, we first letter G if you see it is the 7th letter in the alphabet. So, we say that, ok, I encode 7. So, G represented as 7, becomes my message M, and the ciphertext is obtained as M raised to the power E 7 raised to the power 5 modulo 851, and that gives a 638, just computation and decryption is again M is equal to C raised to the power D modulo N, and we decrypt use in this, and we get back 7. So, even with this moderate, this small numbers we have this kind of a 638 raised to the power 3 one 7 modulo 851.


So, again you can see that the computation is pretty involved.

(Refer Slide Time: 25:16)

Information Theory, Coding and Cryptography

RSA Algorithm

- The RSA algorithm relies on **large prime number**.
- Do we actually have arbitrarily sized prime numbers?
- To answer this question, we first define the **Prime Counting function**.
- **Definition** The **Prime Counting function** $\pi(n)$ counts the number of primes that are less than or equal to n .

 *Indian Institute of Technology, Delhi* 34 *Ranjan Bose
Department of Electrical Engineering*

So, what does RSA algorithm rely on? Relies on large prime numbers, but question again is look if everybody in the world has to use RSA algorithm do we have arbitrarily size prime numbers to begin with. And do we have enough of them or, are we going to repeat this keys we do not want to, we should have as many keys as possible. So, we would like to answer this question about do we have enough large prime numbers, we talk about this prime counting function. The prime counting function π is a function of N , just counts the number of primes that are less than or equal to M , ok, that is the prime counting function.

(Refer Slide Time: 26:01)

Information Theory, Coding and Cryptography

Prime Counting Function

- The table illustrates the prime counting function.

n	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...
$\pi(n)$	1	2	2	3	3	4	4	4	4	5	5	6	6	6	...

- From the table we note that we increment the value of $\pi(n)$ only when we encounter the next prime number.
- The function $\pi(n)$ is monotonically increasing.

Indian Institute of Technology, Delhi 35 Ranjan Bose
Department of Electrical Engineering

Let us say N so, if you look at this time accounting function, you have prime number yes. So, incremented by 1, pi N 3 prime number, yes, 2 4. So, N is 4, but this is no longer prime numbers the count remains 2 and then 5 prime number count increases 6 prime number, no, count does not increase encounter another prime number. So, pi as a function of N increases and so, for this is a prime counting function. So, this function is monotonically increasing, and with that.

(Refer Slide Time: 26:38)

Information Theory, Coding and Cryptography

Enough prime number?

- Let us assume that we have only finite number of prime numbers.
- Let these be p_1, p_2, \dots, p_n .
- Consider the number $m = p_1 \cdot p_2 \dots p_n + 1$.
- Because m is bigger than any prime, it must be a composite number.
- Hence it should be divisible by some prime number.
- However, it is not divisible by p_1 because we get the remainder '1' after dividing m by p_1 .
- Similarly it is not divisible by any $p_i, i = 1, 2, \dots, n$.
- Thus we get a contradiction and hence our assumption was incorrect.
- Therefore we have infinitely many primes.**

Indian Institute of Technology, Delhi 36 Ranjan Bose
Department of Electrical Engineering

So, if you look at the question of are they enough prime numbers, well, let us see whether they are enough or is this infinitely large set. So, let us assume that we only have finite number of prime numbers. And let this set be $P_1 P_2 \dots P_N$. So, we have in this world only N prime numbers that is the assumption. Let us see whether this assumption is right or wrong.


So, let us make another composite number M as a product of this prime numbers plus 1. So, M is larger than any of the prime numbers. And so, since they have only so many prime numbers, right, M is larger than this and M is a composite number. But if it is composite number should be divisible by some prime number. So, that has to have a factor it is a composite number, it is not a prime number. But if you divide by any one of them, because there is are the all known numbers, we have assumed, you always get the residue has 1, the remainder is 1, because it is product of always prime numbers plus 1 which means that it is not a composite number. So, it there is a contradiction, and therefore, we have infinitely many prime number, and that is a saving grace because RSA requires us to have infinite supply of prime numbers.

(Refer Slide Time: 28:09)

Information Theory, Coding and Cryptography

Security of RSA

- The security of RSA algorithm depends on the ability of the hacker to **factorize** numbers.
- New, faster and better methods for factoring numbers are **constantly** being devised.
- The current best for long numbers is the *Number Field Sieve*.
- Prime Numbers of a length that was unimaginable a mere decade ago are now factored easily.
- Obviously the longer number is, the harder it is to factor, and so the better the security of RSA.
- **As theory and computers improve, larger and larger keys will have to be used.**

 Indian Institute of Technology, Delhi

37

Ranjan Bose
Department of Electrical Engineering

What is the security of RSA? Well the security of RSA algorithm depends on the ability of the hacker to factorize numbers, because we have assumed it is easy to multiply difficult to factorize. So, we know that better methods of factorization are coming out the current best is the number field sieve. And of course, on top of add we are getting very


fast computers, which do this factorization much faster. So, it is just race with quantum computing coming in RSA is at a bigger threat. So, as the theory and computer become more efficient larger and larger keys will have to be used.

(Refer Slide Time: 28:56)

Information Theory, Coding and Cryptography

Pretty Good Privacy (PGP)

- **Pretty Good Privacy (PGP)** is a hybrid cryptosystem that was created by Phil Zimmerman and released onto the Internet as a freeware program in 1991.
- PGP's intended use was for **e-mail security**, but there is no reason why the basic principles behind it could not be applied to any type of transmission.
- PGP provides cryptographic privacy and authentication
- **OpenPGP's** encryption can ensure secure delivery of files and messages, as well as provide verification of who created or sent the message using a process called digital signing.
- PGP and its source code is freely available on the Internet, this means that since its creation PGP has been **subjected to an enormous amount of scrutiny by cryptanalysts**, who have yet to find an exploitable fault in it.

 Indian Institute of Technology,
Delhi38Ranjan Bose
Department of Electrical Engineering

We go to now the next so, to say defect standard the pretty good privacy P G. G is a hybrid cryptosystem, right, created in 1991 was released over the internet by Phil Zimmerman, as a freeware program. And it was intended to be used for email security, but it can be used for a variety of applications. PGP provides cryptographic privacy and authentication.


So, both of them are there is very important for email security. I would like to know who sent me that email, whether it is reliable, and it is next [FL] open PGP encryption can ensure secure delivery of files and messages as well as provide verification of who created or send the message using digital signatures. PGP and its source code is freely available on the internet. And this means that has been subjected to a enormous number of tests enormous amount of scrutiny by cryptanalyst, and therefore, it should install a lot of confidence in the users.

(Refer Slide Time: 30:09)

Information Theory, Coding and Cryptography

Elliptic Curve Cryptography

- Most public key cryptosystems get their security from the assumed difficulty of inverting a one-way function.
- **Elliptic Curve Cryptography** (ECC) has become important mainly because groups have been found in which sub-exponential algorithms to invert the discrete exponentiation function are not known to exist.
- Thus one has to use the standard exponential time algorithms to break the security of the conventional public key cryptosystems.
- The basic advantage of elliptic curve cryptosystems is that they are equally secure with smaller key sizes than their conventional counterparts (e.g., RSA).

 *Indian Institute of Technology,
Delhi* 39 *Ranjan Bose
Department of Electrical Engineering*

Let us look at another interesting cryptographic method called the elliptic curve cryptography, or also called as cryptography on the elliptic curve ECC. So, let us see most public key cryptosystems get the security from the assume difficulty of inverting a one-way function, you can multiply 2 a numbers, but not factorize and so on and so forth. ECC has become important mainly, because groups have been found in which sub exponential algorithms to invert the discrete exponential functions are not known to exist. We will talk about this (Refer Time: 30:55) in a short while what do we mean by this discrete logarithmic problem, and discrete exponentiation functions.

So, that is one has to use a standard exponential time algorithms to break the security of the conventional public key cryptosystems. The basic advantage of ECC or the elliptic curve cryptography is that they are equally secure with smaller keys sizes, than then their counterparts like the RSA so, smaller key size, but equal security.

(Refer Slide Time: 31:27)


Information Theory, Coding and Cryptography

Example

- Let Z_p be the set of integers $\{0, 1, 2, \dots, p-1\}$ where p is an odd prime number.
- Let us define an elliptic curve over Z_p as follows

$$y^2 \equiv x^3 + ax + b \pmod{p}$$

with $a, b \in Z_p$ and $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$.

 Indian Institute of Technology, Delhi 40 Ranjan Bose
Department of Electrical Engineering

So, it is kind of an example to understand. Let Z_p be the set of integers 0 1 2 up to p minus 1, where p is an odd prime number, ok. Let us define an elliptic curve over Z_p as follows. So, we are going to define using an example of an elliptic curve y^2 equal to $x^3 + ax + b$, taken modulo p , this is my elliptic curve. And all my calculations multiplications additions will be done over this curve will tell you how.

What are a and b ? a and b also are elements of Z_p . And we put an a constraints this is an example, $4a^3 + 27b^2$ is not equal to 0 modulo of p . So, this is my definition that we are going to use for the elliptic curve and this example.

(Refer Slide Time: 32:26)

Information Theory, Coding and Cryptography

ECC

- For any given a and b in \mathbb{Z}_p , the above equation has a pair of solution x, y in \mathbb{Z}_p which can be expressed as
$$E(\mathbb{Z}_p) = \{(x, y) \mid x, y \in \mathbb{Z}_p \text{ and}$$
$$y^2 \equiv x^3 + ax + b \pmod{p} \text{ and}$$
$$4a^3 + 27b^2 \not\equiv 0 \pmod{p}\}$$
- The resulting set $E(\mathbb{Z}_p)$ consists of all $(x, y) \in \mathbb{Z}_p^2$

41

Indian Institute of Technology, Delhi Ranjan Bose
Department of Electrical Engineering

So, for any a and b , in \mathbb{Z}_p the above equation has a pair of solution x comma y , remember?

(Refer Slide Time: 32:30)

Information Theory, Coding and Cryptography

ECC

- In addition to the points lying on the elliptic curve, one also considers a point O at infinity.
- Some properties of the points on the elliptic curve are listed below.
- $P + O = O + P$ for all $P \in E(\mathbb{Z}_p)$.
- If $P = (x, y) \in E(\mathbb{Z}_p)$, then $(x, y) + (x, -y) = O$.
- The point $(x, -y)$ is also called $-P$.
- Note that $-P$ will always be a point on the elliptic curve.

42

Indian Institute of Technology, Delhi Ranjan Bose
Department of Electrical Engineering

So, we have this x and y here which is defining, so, x comma y is a point on this curve. So, we are now going to talk about points on this curve, defined by this y squared is equal to x cubed plus a x plus b , this some constraints. So, $E \mathbb{Z}_p$ is the tuple x comma y , such that x comma y like on the \mathbb{Z}_p . And x and y are such that this condition is satisfied

y square is equal to x cube plus a x plus b modulo p, and what A and b again we have put this constraint.

So, the resulting set E as a function of Z P consist of all pairs x comma y, element of Z square p. So, basically we have found a set of points, and all we are going to do is work with this set of points. In addition to this points lying on the elliptic curve, we also consider point 0 or O at infinity. So it is a, together they form the complete set of points on the elliptic curve. What are some of the properties? Well, any point P plus O is O plus P for all points on P is right, and if a point P given by x comma y lies on the elliptic curve, then x comma y plus x comma minus y is O. And this x comma minus y is also called minus P, because P minus P where P is a point on the elliptic curve should be O. And what is interesting is if P lies on the elliptic curve, then minus P will always be a point on the elliptic curve we can look at the symmetry.

(Refer Slide Time: 34:41)

Information Theory, Coding and Cryptography


ECC

- Let $P = (x_1, y_1) \in E(\mathbf{Z}_p)$ and $Q = (x_2, y_2) \in E(\mathbf{Z}_p)$ with $P \neq -Q$, then $P + Q = (x_3, y_3)$ where

$$x_3 = \lambda^2 - x_1 - x_2$$

$$y_3 = \lambda(x_1 - x_3) - y_1$$

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{if } P \neq Q \\ \frac{3x_1^2 + a}{2y_1} & \text{if } P = Q \end{cases}$$


43
Ranjan Bose
Department of Electrical Engineering

So, if P is x 1 comma y 1, this kind of a toy example which will bring home the point regarding elliptic curve cryptography. So, let P v B x 1 comma y 1, and Q another point is x 2 comma y 2, right.

Then clearly with P not equal to P Q we have P plus Q as a third point. So, if we add 2 points on the curve, you get a third point also on the curve. And how do you get this addition? Well, that 2 points coordinates x 3 and y 3. X 3 is lambda square minus x 1 minus x 2, and y is equal to lambda x 1 minus x 3 minus x 1, what is lamda? Lamda is

given by this one, provided if P is equal to Q , and P is not equal to Q . So, together we have a rule for finding out addition of 2 points on the curve on the elliptic curve, and sum of 2 points is always on the elliptic curve.

(Refer Slide Time: 35:49)

The slide is titled "ECC" in large blue letters. Above the title, it says "Information Theory, Coding and Cryptography". Below the title, there is a bulleted list of four points. The third point is bolded. At the bottom of the slide, there is a footer with the IIT Delhi logo, the text "Indian Institute of Technology, Delhi", the number "44", and the name "Ranjan Bose, Department of Electrical Engineering".

Information Theory, Coding and Cryptography

ECC

- Thus it is fairly easy to calculate $2P, 3P, \dots, kP$.
- Self addition can also be carried out efficiently using geometric techniques.
- **The Elliptic Curve Cryptography relies on the difficulty of finding k given $Q = kP$.**
- This property will be used in the Diffie-Hellman protocol based on Elliptic Curve.

Indian Institute of Technology, Delhi 44 Ranjan Bose
Department of Electrical Engineering

Now, where does it lead to? When I can always add P 2 to P itself; second get P plus P as $2P$, P plus P plus P $3P$ and so on and so forth to kP .


So, self-addition a multiplication is very, very efficient. So, elliptic curve cryptography relies on the difficulty of finding k given Q is equal to kP . So, this is the basic (Refer Time: 36:21), that is what we are going to do. So, this property will be used in the Diffie-Hellman key exchanged protocol based on the elliptic curve which will go to look at the last example in today's lecture. So, it has the utility is already known of this wonderful mathematical technique.

(Refer Slide Time: 36:40)

Information Theory, Coding and Cryptography

Example

- Let $p = 23$, $a = 1$ and $b = 1$.
- Thus, the elliptic curve can be represented as defined $y^2 \equiv x^3 + x + 1$ over Z_{23} .
- One can verify that indeed $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$.
- The **valid points** on the elliptic curve are O and the following:
(0, 1) (0, 22) (1, 7) (1, 16) (3, 10) (3, 13) (4, 0)
(5, 4) (5, 19) (6, 4) (6, 19) (7, 11) (7, 12) (9, 7)
(9, 16) (11, 3) (11, 20) (12, 4) (12, 19) (13, 7)
(13, 16) (17, 3) (17, 20) (18, 3) (18, 20) (19, 5)
(19, 18)

 Indian Institute of Technology, Delhi 45 Ranjan Bose
Department of Electrical Engineering

Let us look at a little bit more into this elliptic curve cryptography by considering a simple example. We put in numbers is time so, let us say a is equal to 1, b is equal to 1. So, we have this elliptic curve y^2 is equal to $x^3 + x + 1$, ok. And you can verify that $4a^3 + 27b^2$ is not equal to 0, modulo of p, what is p? P in this case is 23 remember we had this Z which had 0 1 2 3 up to odd prime number p.


So, what are the valid point? You can calculate you can plug in through (Refer Time: 37:28) and you can find the set of all tuples, valid points on this curve and they are listed below. So, 0 1 you can substitute and you will find it satisfies equation and so, 0 22 and so on and so forth. So, the all the valid point, as you can see this is a result of a computer such that start with 0 0 then 1 1, then 3 a keep increasing and find the points on this curve, so, that is all it. Take any 2 points at random, add the up and you will get it is a close set will get point on the curve, ok.

(Refer Slide Time: 37:59)

Information Theory, Coding and Cryptography

Example

- (0, 1) (0, 22) (1,7) (1,16) (3, 10) (3, 13) (4, 0) (5, 4) (5, 19)
(6, 4) (6, 19) (7, 11) (7, 12) (9, 7) (9, 16) (11, 3) (11, 20)
(12, 4) (12, 19) (13, 7) (13,16) (17, 3) (17, 20) (18, 3) (18,
20) (19, 5) (19, 18)
- Let $P = (3, 10)$ and $Q = (9, 7)$.
- In order to compute $P + Q$, we have to first determine λ using
$$\lambda = \frac{7-10}{9-3} = \frac{-3}{6} = \frac{-1}{2} = 11 \in \mathbb{Z}_{23}$$
- Consequently,
$$x_3 = 11^2 - 3 - 9 \equiv 17 \pmod{23}$$
$$y_3 = 11(3 - (-6)) - 10 \equiv 20 \pmod{23}$$

 Indian Institute of Technology, Delhi 46 Ranjan Bose
Department of Electrical Engineering

So, these are all the points listed out.

We pick at random P is equal to 3 comma 10, here it is. And take Q is equal to 9 comma 7. So, here it is so, at took 2 points P and Q on the set. And we have interested in finding on P plus Q. So, we had those equations given earlier. So, we found that P is not equal to Q, I use that lambda and then we calculate x 3 and y 3 and calculate mod p, p is 23 here I get 17 come a 20. So, I look in and yes low and low (Refer Time: 28:33) I find 17 comma 20 as a valid quantum (Refer Time: 38:37) So, there is the mathematics over the elliptic curve.

(Refer Slide Time: 38:38)

Information Theory, Coding and Cryptography

Example

- Thus, $P + Q = (17, 20) \in E(\mathbb{Z}_{23})$.
- Next, let us determine $P + P = 2P$.

$$\lambda = \frac{3(3^2)+1}{20} = \frac{5}{20} = \frac{1}{4} = 6 \in \mathbb{Z}_{23}$$
$$x_3 = 6^2 - 3 - 3 \equiv 7 \pmod{23}$$
$$y_3 = 11(3 - 7) - 10 \equiv 12 \pmod{23}$$

- **Thus, $2P = (7, 12) \in E(\mathbb{Z}_{23})$.**

Indian Institute of Technology, Delhi

47

Ranjan Bose
Department of Electrical Engineering

So, this P plus Q 17 comma, 20 is also a point on this curve. And if you want to do P plus P in a ; attempt to find out 2 times P you can calculate the subsequent lambda, and then you can find out x 3 and y 3 as 7 comma 12 and if you go back into the 7 comma 12 is also available.

So, if you add 3 comma 10 with itself, you get 7 comma 12, and so on and so forth I can find out 2 times P 3 P and so on and so forth, and this 2 P also belongs to the set.

(Refer Slide Time: 39:23)

Information Theory, Coding and Cryptography

Example

- Let us look at the Elliptic Curve **Discrete logarithm Problem**.
- Consider the Elliptic curve with $p = 23$, $a = 9$ and $b = 17$.
- Thus, the elliptic curve can be represented as $y^2 = x^3 + 9x + 17$ defined over \mathbb{Z}_{23} .
- We will try to answer the following question: What is the discrete log k of $Q = (4, 5)$ to the base $P = (16, 5)$?
- **That is, how many times should P be added to itself to obtain Q?**
- **The brute force method is to calculate $2P, 3P, \dots$ until kP matches with Q.**
- For example,
- $P = (16, 5)$, $2P = (20, 20)$, $3P = (14, 14)$, $4P = (19, 20)$, $5P = (13, 10)$, $6P = (7, 3)$, $7P = (8, 7)$, $8P = (12, 17)$, $9P = (4, 5)$.
- Thus, $k = 9$.

Indian Institute of Technology, Delhi

48

Ranjan Bose
Department of Electrical Engineering

So, the elliptic curve is a discrete logarithm problem to be considered. So, if you have these elliptic curve that we looked at with P is equal to 23 is equal to 9 and B is equal to 17. So, you have change the A and B from one and one to this one then you have a parallel problem here. And to give you the feel of this discrete logarithm problem we try to answer the following question. What is the discrete log k of Q equal to 4 comma 5 to the base P 16 comma, fine? That is how many times should this P , right? P is a number 16 comma 5, it is a point on the curve.

How many times should P be added to itself to obtain Q ? Like P plus P plus P so, K times P is Q how many times k ? But remember there is a mod in place and that creates the problem. So, the brute force method would be to try first 2 P check whether it is the same as Q , no, then try 3 P check whether it is equal to Q , not equal until K times P whichever matches Q , ok. So, I can try this effort P 2 P , but remember, calculating P and then 2 P and then 3 P require some mathematics, or geometrical you can do on the curve that is an easier one. So, in this example K is 9, this is the discrete logarithm problem.

(Refer Slide Time: 40:56)

Information Theory, Coding and Cryptography

ECC

- In real world k is large, and finding k is computationally very expensive.
- **Elliptic Curve Cryptography** relies on the difficulty of finding k given $Q = kP$.

Indian Institute of Technology, Delhi

49

Ranjan Bose
Department of Electrical Engineering

So, bottom line is a real word K is very, very large and finding k could be computationally very expensive. This is exactly what ECC relies on, the elliptic curve cryptography relies on the difficulty of finding K given Q is equal to kP , that is the basic idea.

(Refer Slide Time: 41:19)

Information Theory, Coding and Cryptography

Diffie-Hellman key agreement protocol

- The protocol allows two users to exchange a secret key over an insecure medium without any prior secrets.
- The protocol has two system parameters p and g .
- They are both public and may be used by all the users in a system.
- Parameter p is a **prime number** and parameter g (usually called a generator) is an integer less than p , with the following property: for every number n between 1 and $p-1$ inclusive, there is a power k of g such that

$$n = g^k \text{ mod } p.$$

Indian Institute of Technology, Delhi 50 Ranjan Bose
Department of Electrical Engineering

Now, last few slides we look at a practical application of this ECC in that form of the Diffie-Hellman key exchange protocol, which allows to uses to exchange a secret key over an insecure medium. So, it is a key exchange (Refer Time: 41:34) agreement protocol, ok. So, it has 2 system parameters p and g p is a prime number and g is called a generator. And we have this n equal to g raised to the power k mod P , will look at a very simple example, to illustrate the point.

(Refer Slide Time: 41:52)

Information Theory, Coding and Cryptography

DH Protocol

- Suppose Alice and Bob want to agree on a shared secret key using the Diffie-Hellman key agreement protocol.
- They carry out the following steps:
- Alice generates a random private value a and Bob generates a random private value b .
- Both a and b are drawn from the set of integers.
- Then they derive their public values using parameters p and g and their private values.
- Alice's public value is $g^a \text{ mod } p$ and Bob's public value is $g^b \text{ mod } p$.
- They then exchange their public values.
- Finally, Alice computes $g^{ab} = (g^b)^a \text{ mod } p$, and Bob computes $g^{ba} = (g^a)^b \text{ mod } p$.
- Since $g^{ab} = g^{ba} = k$, Alice and Bob now have a shared secret key k .

Indian Institute of Technology, Delhi 51 Ranjan Bose
Department of Electrical Engineering

So, what does this DH protocol do? We have these 2 characters in the play Alice and bob, who want to exchange the key?


So, Alice generates a random private value A , and bob generates a random private value B . They are both drawn from the set of integers, then they derive the public value using p and g right, and how do we do that?

(Refer Slide Time: 42:18)

Information Theory, Coding and Cryptography

DH Protocol

- Let us implement the Diffie-Hellman protocol using Elliptic Curve Cryptography.
- **The steps are as follows:**
- Alice and Bob mutually agree on a number P .
- Alice chooses a random private key k_A .
- She then computes $k_A P$ using the elliptic curve and publishes it.
- Bob chooses a random private key k_B .
- He then computes $k_B P$ using the same elliptic curve and publishes it.
- Alice takes $k_B P$ and uses the elliptic curve to calculate $k_A(k_B P)$.
- Bob takes $k_A P$ and uses the elliptic curve to calculate $k_B(k_A P)$.
- Since $k_A(k_B P) = k_B(k_A P)$, Alice and Bob now share a common key.

 Indian Institute of Technology, Delhi

52

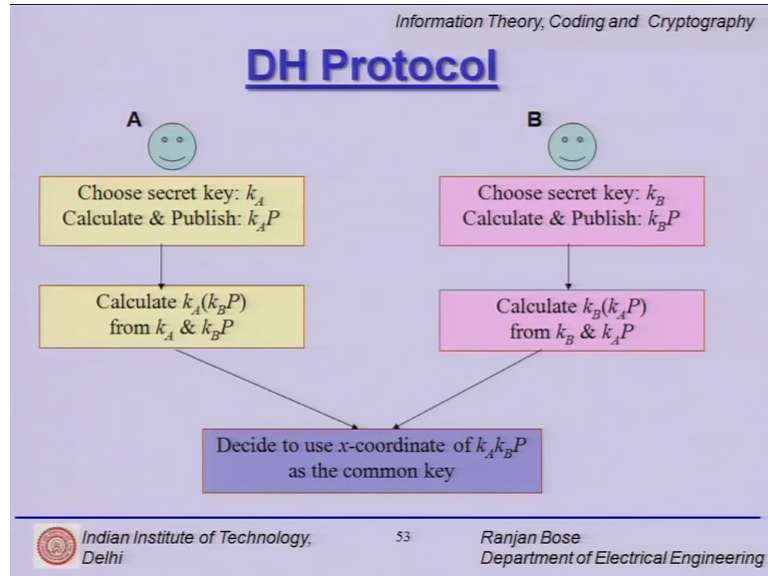
Ranjan Bose
Department of Electrical Engineering

Well this steps are as follows, Alice and bob mutual agree on a number P ok. Alice chooses a private key k_A known only to Alice therefore, k_A is away and then computes $k_A P$. Remember, there is a same time as a point P multiplied with itself with itself so, $k_A P$ using the elliptic curve. And he is publishes $k_A P$, right? But $k_A P$ is not easy, that is the computational difficulty that we have established. Similarly, bob does the same job, he finds chooses at random k_B which is known only to bob, and then computes $k_B P$ using the elliptic curve, same elliptic curve, and then publishes this $k_B P$. Again getting k_B from $k_B P$ is difficult.

Now, what does Alice do? Well, $k_B P$ is known, takes the elliptic curve and again does k_A times $k_B P$, but k_A is known to Alice. What does bob do? Bob knows $k_A P$, it has got it is own k_B known to bob, it does k_B times $k_A P$. But low and behold we have k_A times k_B into P is same as k_B into k_A into P . And magically they have this 2 numbers k_A time k_B into P which are same. So, Alice knows this number, bob knows this number,

but the eavesdropper does not know number. So, Alice and bob have now secretly been able to exchange a key without the eavesdropper knowing it.

(Refer Slide Time: 44:06)



You can graphically represent this. So, Alice chooses k_A calculates $k_A P$ publishes it. Publish means, puts it into the public domain. B bob secretly chooses k_B calculates $k_B P$. Now what is this P ? P is known together, both of them, and known to the outside world as well, so, bob publishes $k_B P$.

And then once the exchange it, then Alice calculates k_A times $k_B P$, bob calculates k_B times $k_A P$, and then they decide to use k_A and k_B as a common k . So, thus they are been able to exchange the k .

(Refer Slide Time: 44:52)

Information Theory, Coding and Cryptography

Summary

- DES
- IDEA
- PGP
- RSA
- DH Protocol

Indian Institute of Technology, Delhi 54 Ranjan Bose
Department of Electrical Engineering

So, with that we come to the end of today's lecture. We have looked at several acronyms, we started with DES which is the precursor of AES. Then we looked at the idea algorithm, we looked at PGP, pretty good privacy followed by the public key methods RSA, and then we looked at the DH Diffie-Hellman key exchange protocol.

With that we come to the end of today's lecture.