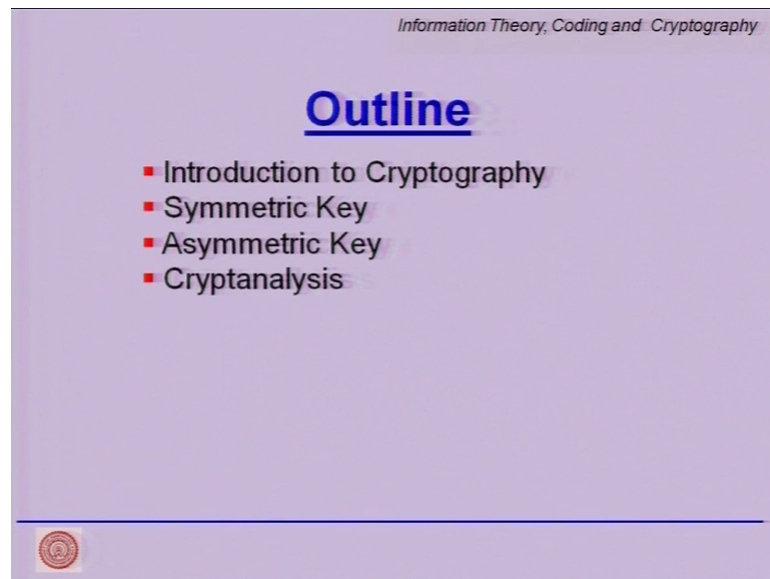


Information Theory, Coding and Cryptography
Dr. Ranjan Bose
Department of Electrical Engineering
Indian Institute of Technology, Delhi

Module – 36
Cryptography
Lecture – 36

(Refer Slide Time: 00:32)




Hello, and welcome to our next lecture, this is on Cryptography. Let us start with the quick outline of today's talk. We would introduce this concept of cryptography followed by asymmetric key cryptography, and then symmetric key followed by asymmetric key cryptography, and finally, we will talk about cryptanalysis. Today's lecture will involve a lot of definitions, because we want to set the ground for understanding the deeper concepts.

(Refer Slide Time: 01:00)

Information Theory, Coding and Cryptography

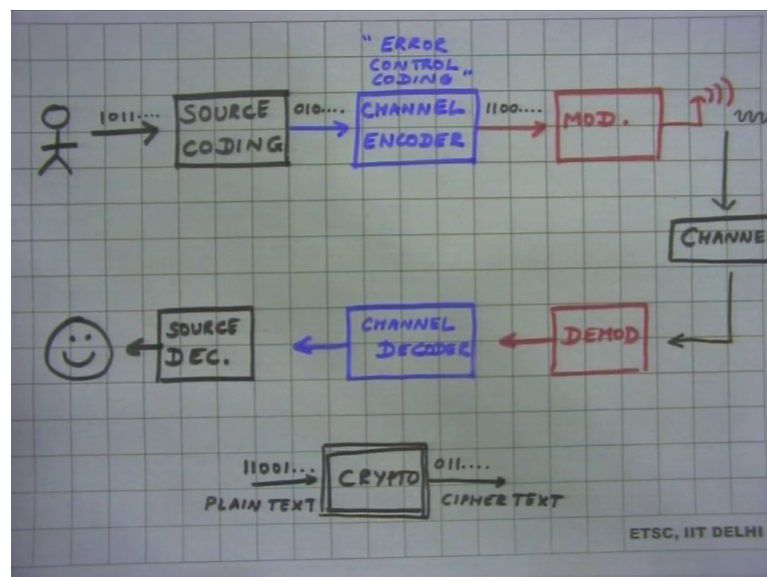
Definitions

- **Cryptography** is the science of devising methods that allow information to be sent in a secure form in such a way that the only person able to retrieve this information is the intended recipient.
- **Encryption** is based on algorithms that scramble information into unreadable or non-discernable form.
- **Decryption** is the process of restoring the scrambled information to its original form.



So, we start with the definitions. The first guy cryptography, well this is the science of devising methods that allow information to be sent in a secure form such that only the person, who is intended to receive right. We call them the intended recipient is able to decipher and retrieve this information. This is in a nutshell what cryptography means ok.

(Refer Slide Time: 01:49)



So, let us have a quick look at our communication system earlier. So, we had a source of information if you see. And then, the first block that we encountered was the source coder. The aim was the source coder was to remove the redundancy from the incoming

bitstrips. Followed the by that we had the channel encoder also will mentioned it, this is the error control coding.

And after this we kind of put the modulation. And then, if it were a wireless communication, were the antenna, which we edited out. At the reverse end, it went through a channel, which introduced noise it could have been a fading channel, it would introduce distortions, delays, and all other kinds of channel imperfection. And then we do the reverse, we have the demodulator followed by the channel decoder, because you have to reverse the things that we have done. And then finally, the source decoding has to happen. And if everything works fine, you get the information back.

Question is nowhere in this loop, have we talked about securing the information, because here you have 1's and 0's coming in. And after source encoding, you have again a series of bits coming out. And post channel encoding we also have bits coming out. And after modulation we have the analogue waveform coming out. Now, these bits need to be protected.

Question is where do you introduce the encoding block, which will be required for cryptographic application, because a person who can intercept this signal, and if we knows, which modulation has been used can demodulate. And if the person the eavesdroppers knows, which channel encoding scheme we have been using, they can do the decoding part, undo the source encoding part, and recover the information, even though, he or she may not be the intended recipient.

So, we have to cut this path open somewhere, and introduce one extra block, the cryptographic block. Now, the cryptographic block, wherever we introduce essentially must take in bits and maybe give out encoded bits. So, there is no way you can get from the cipher text, will define that the plain text. But, the question is where should we introduce this box. Do we cut it and put it here, or because here bits come in bits go out, or do I cut and put it here, bits go in and bits go out, or do I put it right in the beginning, bits go in and bits go out.

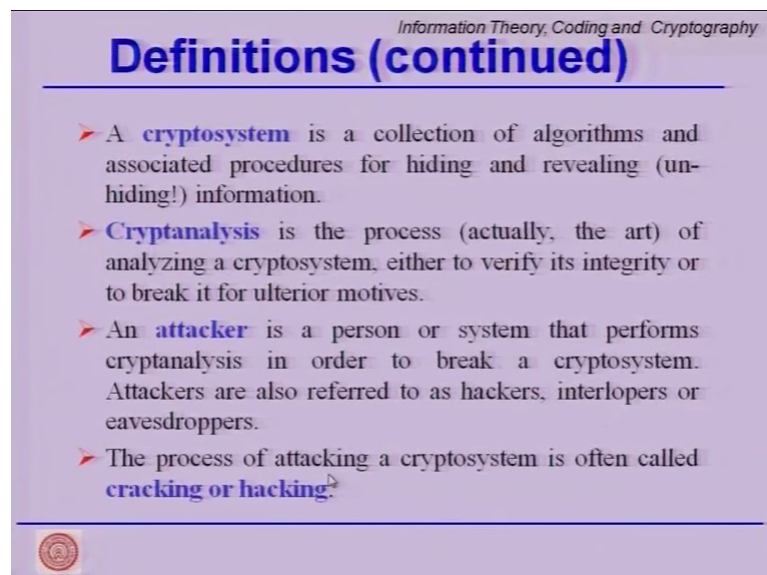
So, what we will realize is that if you insert this box here, then it kind of defeats the purpose of the channel encoding, which is which puts in a certain algebraic structure, so that I can recover from the channel in perfections and noise. So, it is not voice to introduce this crypto block here, because it undoes this job. And if we put it right here,

then we will soon realize that the job of this crypto block is to somehow diffuse information, so make this outgoing bit stream more random in nature ok. And then, if the input to a source coding block is very random, whatever be a degree of randomness be you have a more difficult job for the source encoder to do.

So, the ideal place would be to somehow introduce this box somewhere between these two all right. Otherwise, other than that we can possibly think about ingenious ways is to introduce this box elsewhere, that would be a logical way to insert the box. Once we have this box somewhere in this path, you will have to have a parallel box, the decryption part in the reverse path, so as to get back to happy face to get the correctly decoded output.

So, we go back to our slides, and look at the definitions once more. So, we talk about is encryption, which is nothing but an algorithm running, which kind of scrambles the information to make it unreadable or non-discernable. Decryption is the reverse process of restoring the scrambled information to its original form. So, encryption is done on the (Refer Time: 08:11) by the cryptographic box, and the decryption is similarly done by the crypto block.

(Refer Slide Time: 08:18)



Information Theory, Coding and Cryptography

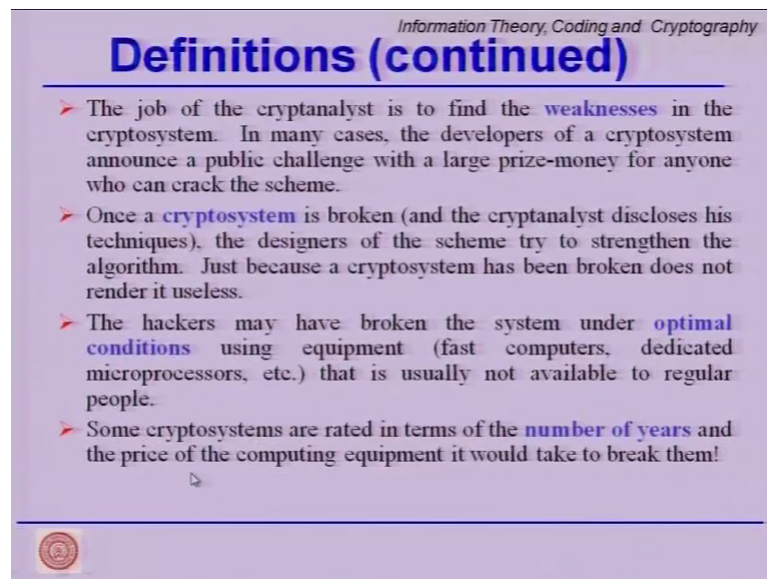
Definitions (continued)

- A **cryptosystem** is a collection of algorithms and associated procedures for hiding and revealing (un-hiding!) information.
- **Cryptanalysis** is the process (actually, the art) of analyzing a cryptosystem, either to verify its integrity or to break it for ulterior motives.
- An **attacker** is a person or system that performs cryptanalysis in order to break a cryptosystem. Attackers are also referred to as hackers, interlopers or eavesdroppers.
- The process of attacking a cryptosystem is often called **cracking or hacking**.

So, what is the cryptosystem? Well it is a collection of algorithms and associated procedures for hiding and revealing the information. Cryptanalysis is a word, will use later in this lecture, is a process some call it an art of analyzing a cryptosystem, because

there is no one way to analyze a cryptosystem ok. But, what do we analyze, we find out the integrity how difficult it is to break into the cryptosystem. An attacker is a person or a system that performs cryptanalysis in order to break a cryptosystem, we have different words and synonyms for an attacker, hackers interlopers, eavesdroppers are all used in literature. The process of attacking a cryptosystem is called cracking or hacking.

(Refer Slide Time: 09:12)



Information Theory, Coding and Cryptography

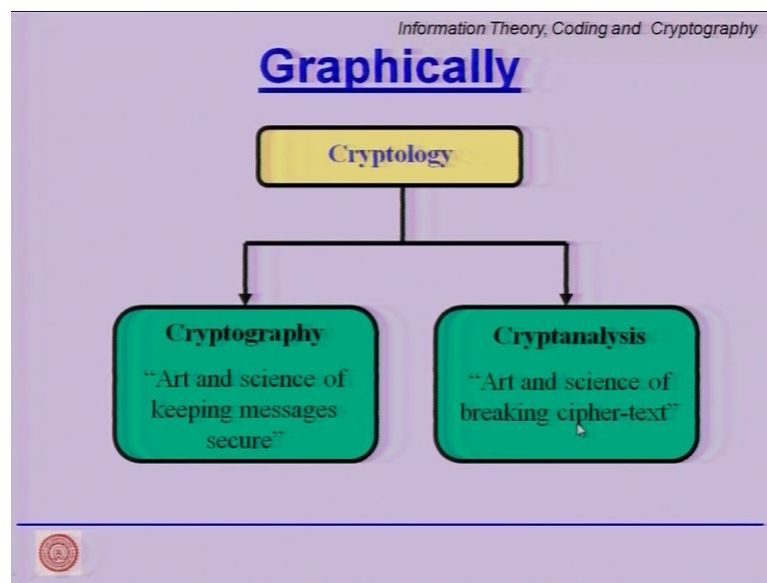
Definitions (continued)

- The job of the cryptanalyst is to find the **weaknesses** in the cryptosystem. In many cases, the developers of a cryptosystem announce a public challenge with a large prize-money for anyone who can crack the scheme.
- Once a **cryptosystem** is broken (and the cryptanalyst discloses his techniques), the designers of the scheme try to strengthen the algorithm. Just because a cryptosystem has been broken does not render it useless.
- The hackers may have broken the system under **optimal conditions** using equipment (fast computers, dedicated microprocessors, etc.) that is usually not available to regular people.
- Some cryptosystems are rated in terms of the **number of years** and the price of the computing equipment it would take to break them!

Somehow this terms we have come across in different context. So, a cryptanalysis is a good guy, still trying to break the cryptosystem, but in the good sense is trying to look at the weaknesses in the cryptosystem. Once the cryptosystem is broken, the cryptanalyst discloses the techniques, and the designers go back on the drawing board, and try to figure out how to make the cryptosystem strongest, so it is a cat and mouse game. So, hackers possibly could have broken the system under optimal conditions using super fast computer, dedicated microprocessor, network, distributed computing etcetera, which are not available to common people. So, we must keep that in mind regarding the strength of an algorithm.

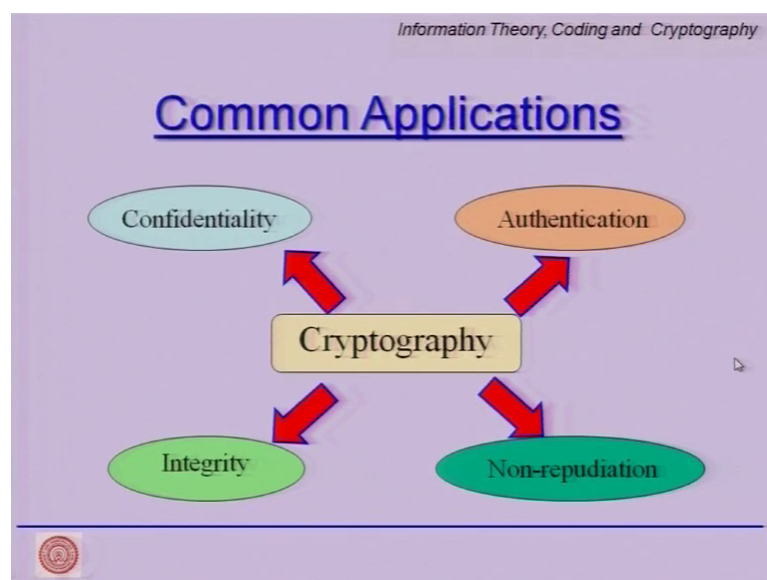
So, some cryptosystems are rated in terms of numbers of years it takes to a break into them, and the price of the computing equipment available. So, these two together decide how strong the cryptosystem is. So, essential it boils down to the resources available to the adversary.

(Refer Slide Time: 10:20)



So, let us look at this whole domain of cryptology. So, how does cryptology relate to cryptography. So, let us draw the binary tree. So, cryptology has two aspects, cryptography and cryptanalysis, together this is the cryptology. Cryptography is the art and science of keeping the message secure, whereas cryptanalysis is the art and science of breaking the ciphertext. Cipher-text is the encoded message.

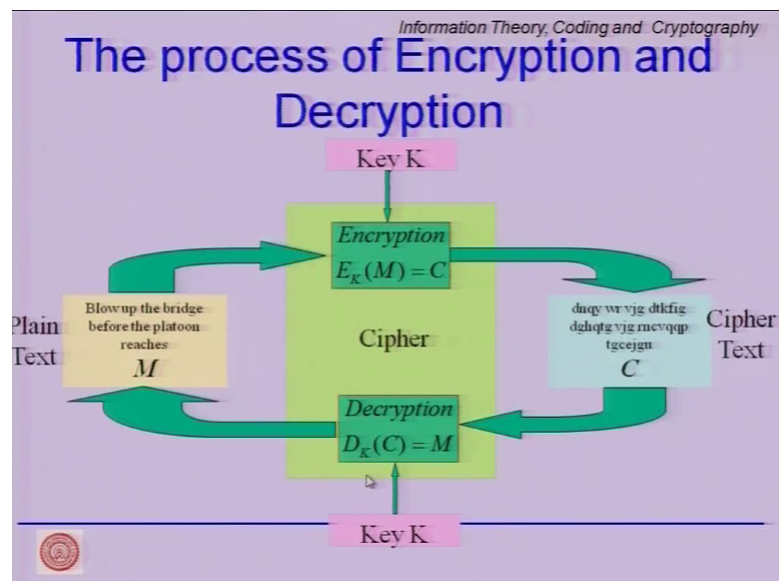
(Refer Slide Time: 10:54)



Before we proceed any further the common areas utility for cryptography are well foremost is confidentiality. Then I would like to authenticate the person is sending the

information is very important for me, where the information is coming from; Integrity of the information, so nobody has tampered with it or change the content during transmission. And non-repudiation the sender cannot really deny that the person has sent this information. So, all of this will become very critical in some of the very practical application such as digital banking, mobile money, so digital signatures all of that legalese will come into picture, once we talk about real life applications.

(Refer Slide Time: 11:48)

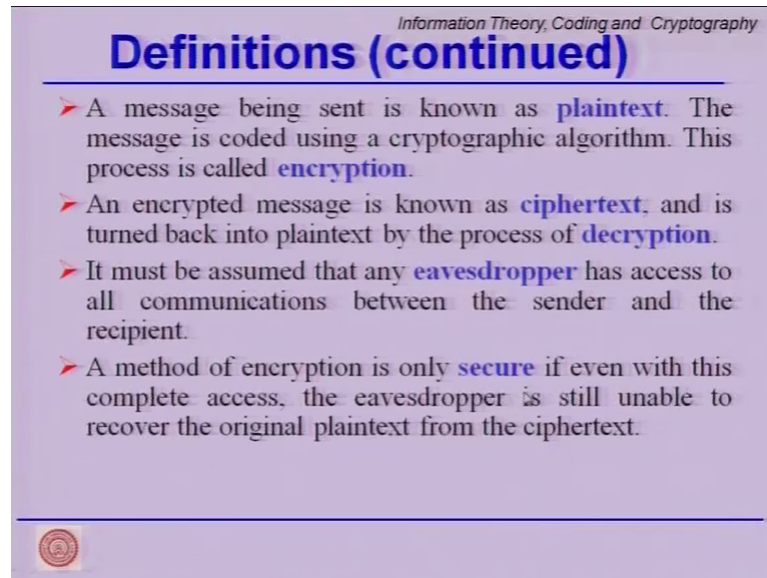


So, let us quickly see how this encryption and decryption works. We always start with a plain text. Plain text is the uncoded simple message, so for example, we want to transmit blow up the bridge before the platoon reaches, this is denoted by M , my message plain text they all use interchangeably. So, this message goes through a box called encryption. This encryption algorithm E or it could be a collection of algorithms, takes in this message M , uses K , which is the key all right. We will talk about what is the key, and generates a cipher text C .

So, C might look like some garbled information junk this is the cipher text, and it should really did not make much sense; Unless of course, you have the description algorithm, and if this is asymmetric algorithm, which we will discuss in detail. Then the decryption algorithm could be another collection of algorithms, which uses the same key K takes into in as an input C , the cipher text and restores it back to the original plain text M . So, this is the process of encryption and decryption.

In this particular brand of cryptography, where using the same key K for encryption and decryption, and therefore, we might say that this is a symmetric key cryptosystem. Now, please note that this key must be kept secret, because the moment the adversary or the eavesdropper has this key, he or she will be able to decrypt the message. So, sometimes this is also referred to as secret key encryption.

(Refer Slide Time: 13:50)



The slide is titled "Definitions (continued)" in a large, bold, blue font. Above the title, in a smaller font, is "Information Theory, Coding and Cryptography". Below the title, there are four bullet points, each preceded by a red arrow. The text in the bullet points is as follows:

- A message being sent is known as **plaintext**. The message is coded using a cryptographic algorithm. This process is called **encryption**.
- An encrypted message is known as **ciphertext**, and is turned back into plaintext by the process of **decryption**.
- It must be assumed that any **eavesdropper** has access to all communications between the sender and the recipient.
- A method of encryption is only **secure** if even with this complete access, the eavesdropper is still unable to recover the original plaintext from the ciphertext.

At the bottom left of the slide, there is a small circular logo.

So, message being sent is known as a plain text right. And the algorithm is used to encrypt it, and there are by we get the cipher text. Coming back to plain text on the ciphertext is called decryption. And we assume that the eavesdropper has access to all communication between sender and receiver, so we do not believe in hiding stuff. We assume that eavesdropper will by hook or by crook get all the communication between the sender and receiver, and then try to crack the code.


So, the method of encryption is only secure, if even if this is completely accessible to the eavesdropper. And still the eavesdropper cannot recover the image, despite knowing what algorithm is being used, what is the process, and all the communication between the transmitter and receiver.

(Refer Slide Time: 14:44)

Information Theory, Coding and Cryptography

Definitions (continued)

- A **key** is a value that causes a cryptographic algorithm to run in a specific manner and produce a specific ciphertext as an output.
- The **key size** is usually measured in bits.
- The bigger the key size, the more secure will be the algorithm.




Now, we come to the key. Key is a value ok, it is a number it could be a chain of bits, it could be a character, it could be a pin number. A key is a value that causes a cryptographic algorithm to run in a specific manner, and produce a specific ciphertext as an output that is the job of a key. Now, the question is how big should the key be. Well the key size is usually measured in bits, and the intuition is that bigger the key size, more secure will be the algorithm.

(Refer Slide Time: 15:28)

Information Theory, Coding and Cryptography

Example

- Suppose we have to encrypt and send the following stream of binary data (which might be originating from voice, video, text or any other source)
 $0110001010011111\dots$
- We can use a 4-bit long key, $x = 1011$, to encrypt this bit stream. To perform encryption, the plaintext (binary bit stream) is first subdivided into blocks of 4 bits.
 $0110 \ 0010 \ 1001 \ 1111\dots$
- Each sub-block is XORed (binary addition) with the key, $x = 1011$. The encrypted message will be:
 $1101 \ 1001 \ 0010 \ 0100\dots$
- The recipient must also possess the knowledge of the key in order to decrypt the message. The decryption process is fairly simple in this case. The ciphertext (the received binary bit stream) is first subdivided into blocks of 4 bits. Each sub-block is XORed with the key, $x = 1011$. The decrypted message will be the original plaintext
 $0110 \ 0010 \ 1001 \ 1111\dots$
- It should be noted that just one key is used both for encryption and decryption.



So, let us look at a very simple example. We have a bit stream coming here, 0 1 1 0 0 0. so and so forth. And we wish to in encoded using a 4-bit long key, so the key length is 4 bits, and my key x is 1 0 1 1. So, what we do is, we take the input bit stream, first break it up in to blocks of the key length, key length is 4 bits. So, as we can see inserts spaces, and break up the input bit stream into sub-blocks, each one is of the length of the key. Now, we take one blocker a time and exhort it with the key right. So, once you do exhorting you will change this four bits depending on the key. So, this is the encoded bit stream, and if you compare it to the 1, which we received as the plain text it is different, and key has caused it to be different.


Now, if the recipient also knows the same key, then he or she can do in exhorting process once again, and immediately will get back the original plain text, because key exhort with itself will give me all 0s and therefore, it tantamounts to exhorting it with the string of 0s. And hence, the original message is retrieved back. And here is the output, once we exhort it back again, and if you see this is the original plain text image. So, this is a very simple conceptual example, which tells us how a key may be used possibly to encode your input bit stream. Please note that here, we have use the same key for encoding and decoding.

(Refer Slide Time: 17:27)

Information Theory, Coding and Cryptography

Example

- Let us devise an algorithm for text messages, which we shall call *character + x*. Let $x = 5$. In this encryption technique, we replace every alphabet by the fifth one following it, i.e., A becomes F, B becomes G, C becomes H, and so on.
- The recipients of the encrypted message just need to know the value of the key, x , in order to decipher the message. The key must be kept separate from the encrypted message being sent.
- Because there is just one key which is used for encryption and decryption, this kind of technique is called **symmetric cryptography** or **single key cryptography** or **secret key cryptography**.

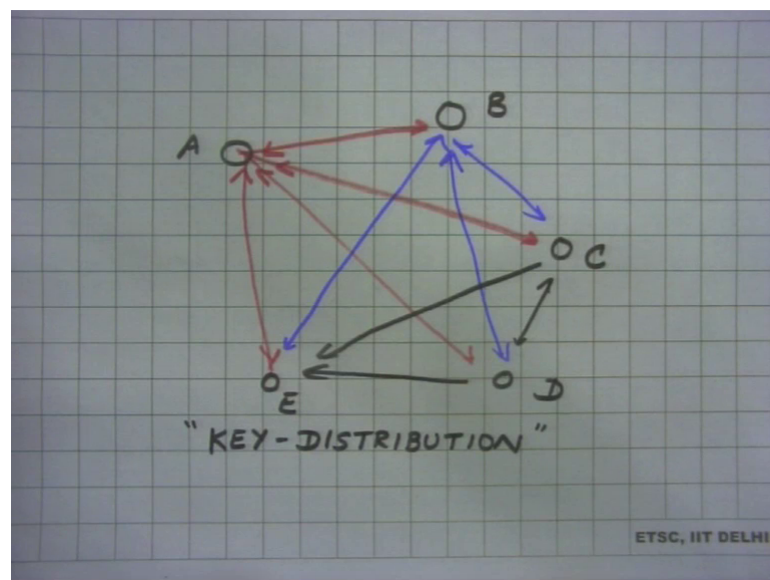


Let us take an another example, which is slightly different. Here, the key is not in bits, key is a number. Let us say x is my key, and my algorithm is character plus x . So, let us

say x, and I am trying to encoding English alphabet. So, A is substituted by the 5th alphabet, because x is 5 B is substituted by the 5th alphabet following it, so B becomes G, C becomes H, and so and so forth. So, a person who has to decode must know this value of this key, in order to decrypt the message.

But, please note, all these examples required the input key for encryption and decryption to be the same and therefore, it is called the single key cryptography, or symmetric key cryptography, and the need to keep the key secret gives its the name the secret key cryptography; So, they all the same, symmetric key, single key, and secret key cryptography.

(Refer Slide Time: 18:51)



This is as a post to public key. Now, the reason is as follows. Let us look at a very simple example as a motivation for public key encoding. So, if you go back to the drawing board, and look at notes; so, suppose we have a network, which has several nodes, and they all need to communicate to each other. And suppose it is a wireless networks, so everybody else can hear, everybody's communication.

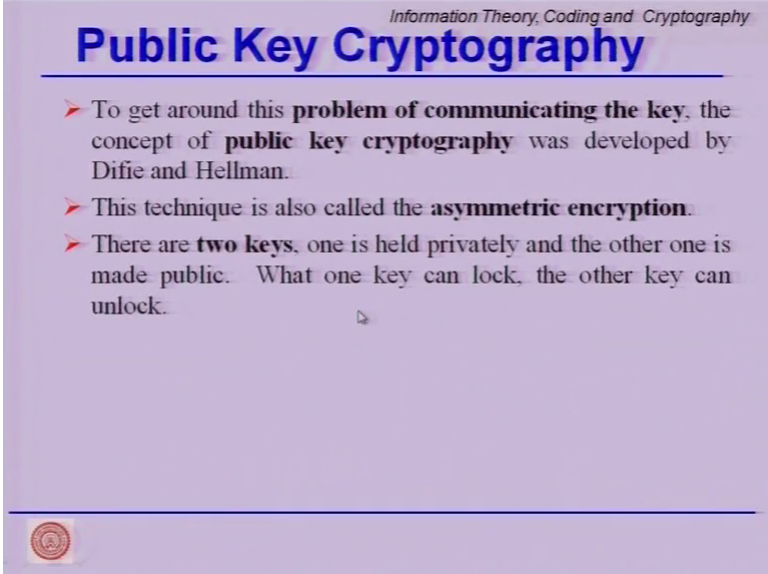
So, A needs to talk to B also to C to D and to E, and the communication could be two way. Similarly, B has to talk to A to E to C to D. So, you can go ahead, and keep drawing this interconnections, and to very soon you will see that this is pretty connected a network. Now, for each pair we need to have one secret key, but it is not a question of

just having the secret key that key must have been sent across. So, key is need to be distributed, this brings us to the problem of key distribution.

So, it is easy to calculate that for this very simple wireless network. I need to have several pairs of key to work, and they have to be distributed right in the beginning before any communication starts. But, this key has to be distributed through a secure channel or a smart way.

Otherwise, who knows if I send a key across, the key is intercepted, and then all by remaining communication is compromised. So, this key distribution will become a major concern, unless we solve this problem. One of the ways to solve this problem is to have the notion of public key cryptography or asymmetric keys; so, that you can easily transport the keys right in the beginning without resorting to a secret key algorithm.

(Refer Slide Time: 21:08)



Information Theory, Coding and Cryptography

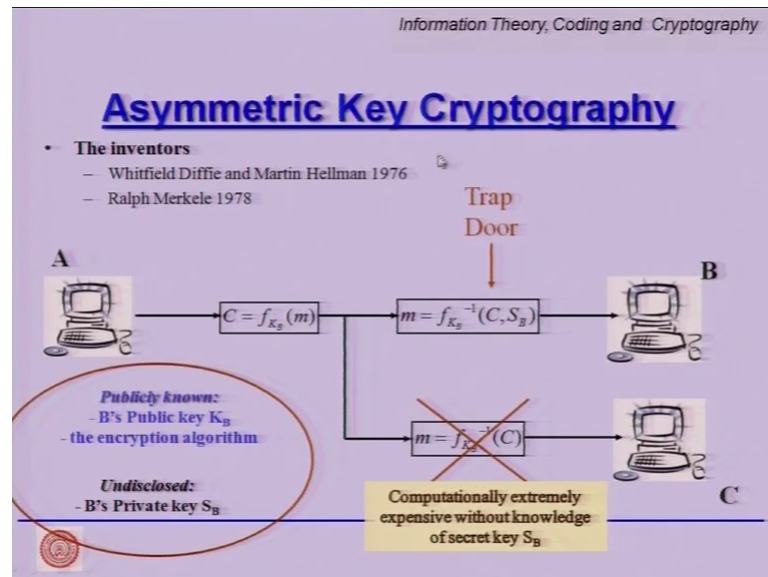
Public Key Cryptography

- To get around this **problem of communicating the key**, the concept of **public key cryptography** was developed by Diffie and Hellman.
- This technique is also called the **asymmetric encryption**.
- There are **two keys**, one is held privately and the other one is made public. What one key can lock, the other key can unlock.

So, we go back to the slides, and look at this public key cryptography. As I said the problem of communicating the keys is difficult, and to get around the problem, the public key cryptography was develop by Diffie and Hellman. And this is also called asymmetric keys encryption or asymmetric encryption, because the key required to encode is not the same as the key required to decode.

Thus there are two keys, one is held privately is called the private key, the other one is made public and is called the public key. What one key can lock, the other key can unlock. So, it is like two interlocked padlocks, and you can open either one.

(Refer Slide Time: 21:58)



So, let us look at conceptually what is happening. And the basic idea is suppose A or Alice wants to communicate to B is give a name Bob. Then this message M is passed through algorithm f , which uses the K_B . K_B could be the public key or B's public key right, this is known or published in advance.

So, if A wants to communicate to B, it grabs the public key of B or downloads it, it is available openly. Uses that to encode the plain text message, and make it C. Now, C B has a private key or secret key S_B that in conjunction with the cipher text, you can have another decryption algorithm, which will give you back the plain text image.


So, please note, there are two keys, public key of B, K_B , and private key S_B . The interesting part is that this algorithm is very easy to conduct in the forward direction, but extremely difficult to conduct in the backward direction. So, it's computationally extremely expensive without the knowledge of the secret key S_B to recover the message from the cipher text ok; so, it's reliance on computational complexity required without the trap door.

(Refer Slide Time: 23:57)

Information Theory, Coding and Cryptography

Illustration

- Suppose we want to send an encrypted message to recipient A using the public key encryption technique.
- To do so we will use the public key of the recipient A and use it to encrypt the message.
- After the recipient receives the message, recipient A decrypts it with his private key.
- Only the private key of recipient A can decrypt a message that has been encrypted with his public key.
- Similarly, recipient B can only decrypt a message that has been encrypted with his public key.
- Thus, no private key ever needs to be communicated and hence one does not have to trust any communication channel to convey the keys.



So, suppose you want to send an encrypted message to recipient A, using the public key encryption technique. So, you will use the public key of the recipient A, to use it to encrypt the message. And A will then take that ciphertext, and use his private key to decrypt. And clearly, only the private key of A can decrypt, so nobody else, even though you openly send it out on all frequencies, only A with his private key can decrypt the message that has been encrypted with his public key, so that makes the algorithm secure.


Similarly, recipient B can only decrypt a message that has been encrypted with his or her public key. Thus no private key ever needs to be communicated. (Refer Time: 24:45) initially generate a pair, private key and public key, publisher, public key, and retain your private key, thus the key distribution problem has been solved.

(Refer Slide Time: 25:18)

Information Theory, Coding and Cryptography

Digital Signatures

- Suppose we want to send somebody a message and also provide a proof that the message is actually from us (a lot of harm can be done by providing bogus information, or rather, misinformation!).
- In order to keep a message private and also provide the authentication (that it is indeed from us), we can perform a special encryption on the plain text with our *private* key, then encrypt it again with the *public* key of the recipient.
- The recipient uses his private key to open the message and then use our public key to verify the authenticity. This technique is said to use **digital signatures**.



Let us look at a very interesting application the digital signatures, which sooner or later is going to invade our lives. Every transaction that we carry out over the internet, may have to be signed with digital signatures for our own safety, especially the banking staff, and the legal staff over the internet.

Suppose we want to send somebody a message, and also provide a proof that the message is actually from us. Why? Why is this important, it was a lot of harm can be done, but either providing bogus information, or so to say misinformation. Plus similar to the physical signature, we would like to have a digital signature that will link the signature to me all right, so that is what we would like to create digitally.


So, in order to keep a message private, and also provide the authentication, we can perform a special encryption on the plain text with a private key, and then encrypted again with the public key of the recipient. The recipient uses his private key to open the message, and then use our public key to verify the authenticity. This technique is called digital signatures.

(Refer Slide Time: 26:24)

Information Theory, Coding and Cryptography

One-Way Functions

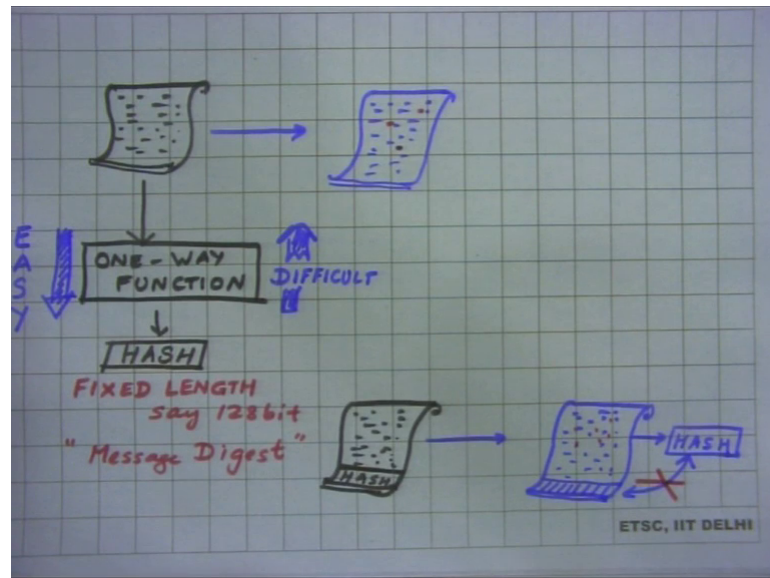
- Another important encryption technique called the **one-way function**.
- It is basically a non-reversible quick encryption method. The encryption is easy and fast, but the decryption is not.
- Suppose we send a document to recipient A and want to check at a later time whether the document has been tampered with.
- We can do so by running a one-way function, which produces a fixed length value called a **hash** (also called the **message digest**).
- The hash is the unique signature of the document that can be sent along with the document.



Now, let us talk about one-way functions. Another interesting technique is called one-way functions, which will help us create the hash values. What is it, it is basically a non-reversible quick encryption method. Quick is a keyword, it is extremely fast, encryption is easy and fast, but the decryption is not. So, it is a one-way thing.

Suppose we want to send a document to recipient A, and want to check at the later time, whether the document has been tampered with ok; so, we would like to look at the integrity. So, we can do so by running a one-way function, which produces a fixed length value called a hash, also called as a message digest. The hash is the unique signature of the document that can be sent along the document.

(Refer Slide Time: 27:22)



So, let us look at it, a little bit more graphically. Suppose, we have to send out a long message, it could be my will, it could be a legal document, it could be my bank statement, so this document has to be transported. But, it is possible that in the process of sending somebody has tampered with this long document, such that at some places, some errors have been introduced or some modifications have been done. For example, there are lot of numbers in my will, and a few 0's have been added or deleted here and there, in this forty page document, and have no means to check whether it is tampered with or not, because this is digital, and this was digital as well.

So, I would like to have a way to authenticate this. So, one way to do it, is use a one-way function. This one-way function as the name suggest takes this long document, and converts it into a hash value. Now, this hash is a fixed length hash, this is important. So, regardless of the size of the document, the hash value is fixed length say 128 bits, so it is a 128 bit hash. So, whether it is a forty page document, or a five page document, at a one small image, I get the fixed length hash, and it is also called as a message digest in some books.

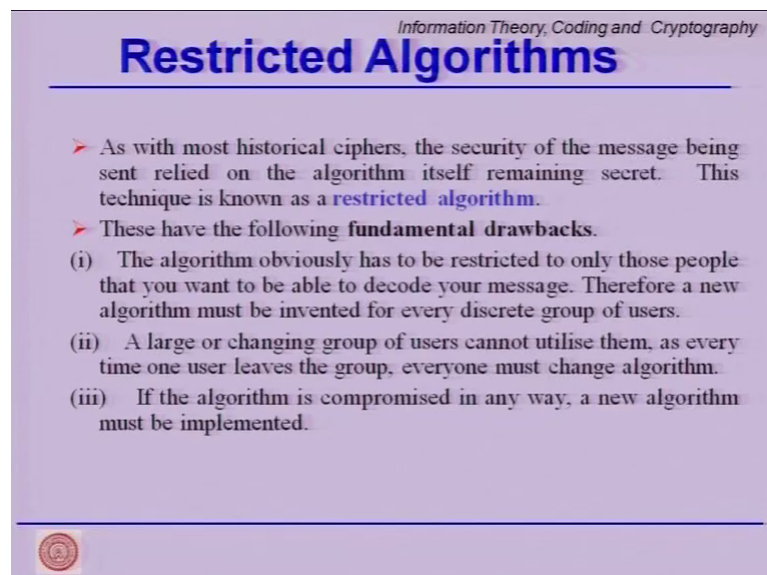
So, what I do now is I append this hash value to the message. So, what I do is take my original message, and before sending at the end I attach my hash value and transmitter. So, I know exactly, how much is the message, and a hash value.

Now, if the message is not tampered with at the receiver, then when I do a hash calculation again, and it is a very fast efficient one-way function, it is very easy to go. So, this is this way is easy, is extremely easy to take a message and find out its hash. But, it is very difficult, to go the other way round, it is very difficult to go, and hence the name one-way hash function. So, I can quickly compute the hash value.

So, what I do is the moment I receive this information, I compute the hash value, and I compare with the hash value attached, if it matches, I jump for joy and say, my message has not been tampered with. With the beauty of this one-way functions is even if a few a 1 or 2 bits have been tampered with changed, modified, then the hash value will not match all right. So, there is a whole body of research work, which has come out with very good hash functions, because clashes are lightly, because big documents are converted into fixed length hashes. So, it is possible that two different documents may give the same hash value. But, these clashes, this collisions are not very lightly.

So, we come back to our one-way function, and we talk about this process that how a recipient can run the one-way function immediately, and check the document has been altered or not.

(Refer Slide Time: 32:29)



Information Theory, Coding and Cryptography

Restricted Algorithms

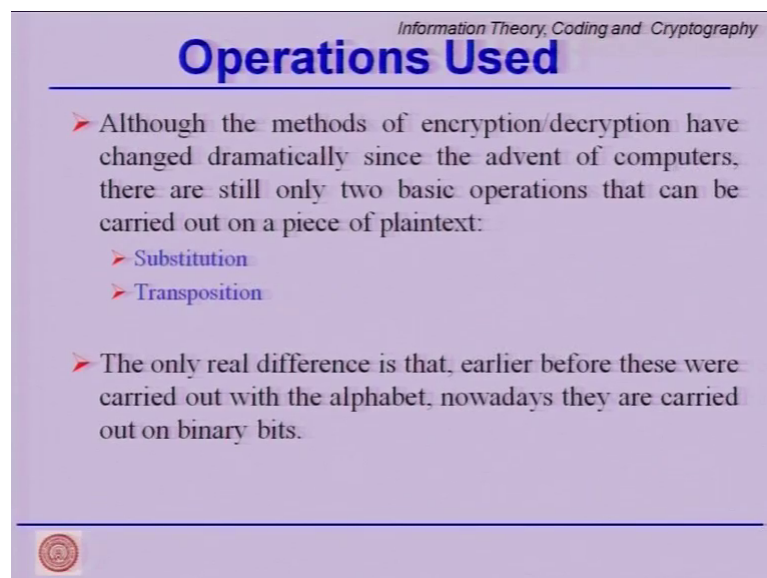
- As with most historical ciphers, the security of the message being sent relied on the algorithm itself remaining secret. This technique is known as a **restricted algorithm**.
- These have the following **fundamental drawbacks**.
 - (i) The algorithm obviously has to be restricted to only those people that you want to be able to decode your message. Therefore a new algorithm must be invented for every discrete group of users.
 - (ii) A large or changing group of users cannot utilise them, as every time one user leaves the group, everyone must change algorithm.
 - (iii) If the algorithm is compromised in any way, a new algorithm must be implemented.

Now, we comeback to restricted algorithms so, this is for the sake of completeness. As with most historical cipher, security of message being sent relied on the algorithm itself remaining secret. So, earlier the algorithm was not disclosed, and this was the domain of

restricted algorithm, this method is still used in the defense, but in general for all several applications we rarely use restricted algorithms, because the restricted algorithms have the following feedback the drawbacks.

The algorithm obviously, has to be restricted to only those people that you want to be able to decode your message, but they need to know. So, therefore, a new algorithm has to be invented for every discrete group of users. So, it is practically difficult, unless you are in a different scenario. A larger changing group of users cannot utilize them, as every time a user leaves, you must change the algorithm, because the cat is out of the bag. And number three, if the algorithm is compromised in anyway, a new algorithm must be implemented.

(Refer Slide Time: 33:51)



Information Theory, Coding and Cryptography

Operations Used

- Although the methods of encryption/decryption have changed dramatically since the advent of computers, there are still only two basic operations that can be carried out on a piece of plaintext:
 - Substitution
 - Transposition
- The only real difference is that, earlier before these were carried out with the alphabet, nowadays they are carried out on binary bits.


So, now let us talk about the general operations that I used for encoding encryption, in terms of the cryptographic algorithms. So, the two most common operations are, substitution and transposition. Substitution you basically replace one character by another character. Transposition means, you change the position right. Earlier, classically cryptography was done alphabet wise, but now most of the algorithm work on bits.

(Refer Slide Time: 34:20)

Information Theory, Coding and Cryptography

Substitution

- Substitution operations replace bits in the plaintext with other bits decided upon by the algorithm, to produce ciphertext.
- This substitution then just has to be reversed to produce plaintext from ciphertext.
- **Example** Julius Caesar was one of the first to use substitution encryption to send messages to troops during the war. The substitution method he invented advances each character three spaces in the alphabet. Thus,
 - THIS IS SUBSTITUTION CIPHERbecomes
 - WKLV LU VXEVLWXWLRQ FLSKHU.




So, substitution is basically you replace bits in the plaintext with other bits decided by the algorithm. Classical example is the Julius Caesar cipher, also called as the Ceaser's cipher, where you substitute one letter for the other. So, T is replaced by W, H is replaced by K. So, you can have a look-up table, and decide what the substitution as, so this becomes totally intelligible.

(Refer Slide Time: 34:54)

Information Theory, Coding and Cryptography

Transposition

- Transposition (or permutation) does not alter any of the bits in plaintext, but instead move their positions around within it.
- If the resultant ciphertext is then put through more transpositions, the end result is increasingly secure.
- **Example:** Use of Permutation Matrix



Transposition basically is permutation, it does not alter the bits or the character, but just moves the position around. So, it is like a scrambling of sorts, you can use the permutation matrix to very efficiently scramble.

(Refer Slide Time: 35:08)

Information Theory, Coding and Cryptography

Shannon's Principle of Confusion

Substitution Cipher

all to attack simultaneously at dawn

Caesar's Cipher

4 cyclic shifts

eppxs exxeg owmqy pxeri sywpc exhea r

26! Possible keys

mzzdu mddmn lypax zdmfq uxzyt mdjmv f

So, Shannon also made some contribution. And this is Shannon's principle of confusion, in terms of the substitution cipher, where different characters are being replaced by different other characters, and you can do a substitution cipher also known as the Ceaser's cipher.

(Refer Slide Time: 35:32)

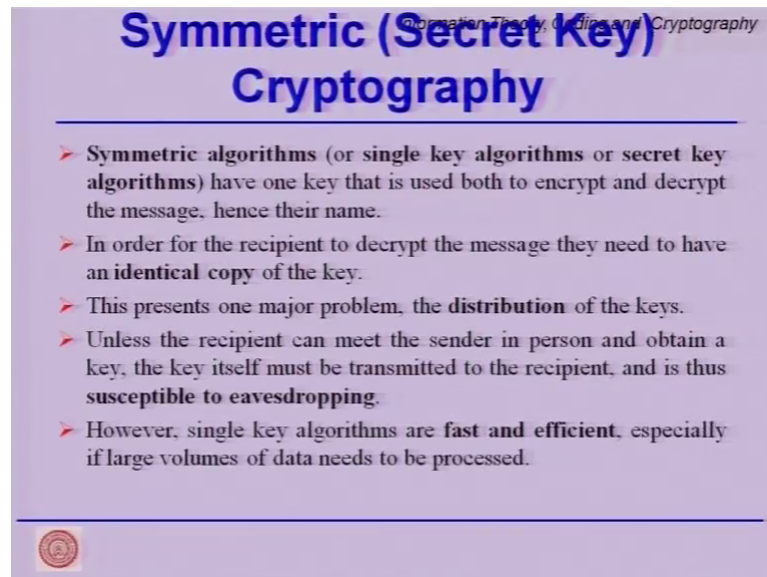
Information Theory, Coding and Cryptography

Entropy of English language

- Single character statistics
 - Entropy $H = 4$ bits/character
- Written english taking into account the full context
 - Shannon (1950): Entropy $H = 0.6$ to 1.3 bits/character
 - Simulations (1999): Entropy $H = 1.1$ bits/character
- **Compression before encryption increases security**
 - Good data compression algorithms (eg. Lempel-Ziv) remove all redundancy and come very close to entropy of the plain text.

So, we are talked about entropy of English write in the beginning, and you can see that once to do either a substitution or a transposition you tend to increase the entropy, so that it tantamounts should randomizing the plaintext. Therefore, we should not use an cryptographic algorithm before the source coding block.

(Refer Slide Time: 36:08)



Symmetric (Secret Key) Cryptography

- Symmetric algorithms (or single key algorithms or secret key algorithms) have one key that is used both to encrypt and decrypt the message, hence their name.
- In order for the recipient to decrypt the message they need to have an **identical copy** of the key.
- This presents one major problem, the **distribution** of the keys.
- Unless the recipient can meet the sender in person and obtain a key, the key itself must be transmitted to the recipient, and is thus **susceptible to eavesdropping**.
- However, single key algorithms are **fast and efficient**, especially if large volumes of data needs to be processed.

Now, let us quickly revisit the symmetric key cryptography or the secret key algorithms. Where we use a single key, which has to be used for encoding and decoding, and we have mentioned that the decoder must also have an identical copy of the thing. And the biggest problem we talked about was distribution of keys.


And unless the recipient can meet the sender in person or get a secret channel, or a secure channel the key, or sending of the key is susceptible to be eavesdropped. So, the advantage is single key algorithms are extremely fast and efficient, so whenever we have to do large volumes of data, we must go for the symmetric key cryptosystems.

(Refer Slide Time: 36:54)

Information Theory, Coding and Cryptography

Size of the key?

- The size of the key is critical in producing strong ciphertext. The U.S. National Security Agency NSA stated in the mid-1990s that a 40-bit length was acceptable to them (i.e. they could crack it sufficiently quickly!). Increasing processor speeds, combined with loosely-coupled multi-processor configurations, have brought the ability to crack such short keys within the reach of possible hackers.




There is a question is how big the key should be. So, we agree that the size of the keys critical in producing strong ciphertext. This is a reason why, we have always prompt at use large passwords to login, because that is kind of a key for us right. But, increasing processor speed, combined with multi-processor system, distributed computing, we have really a problem with the size of the key. So, today most schemes use 1024-bit keys, or 2048-bit or even longer keys, those are the typical size.

(Refer Slide Time: 37:36)

Information Theory, Coding and Cryptography

How long should the key be?

- There is no single answer to this question.
- It depends on the specific situation.
- To determine how much security one needs, the following questions must be answered:
 1. How much is the data to be protected worth?
 2. How long does it need to be secure?
 3. What are the resources available to the cryptanalyst/hacker?



But the question is how long should the key B, and the answer is there is no real one answer to the question, it depends on the circumstances, and the data to be protected. So, question we must ask is how much security does one need for the particular data right. First question, how much is the data to be protected worth, is it worth? Investing in those many keys in the cipher system.


And the second most important thing is how long does the data need to be secure? Is it tomorrow's headline that we are protecting, or a diplomatic secret that has to be protected for 50 years, so how long does the data need to be secure. And the third question is what are the resources available to the cryptanalyst or the hacker, because that will decide how long will it take for the adversary to break into the system.

(Refer Slide Time: 38:36)

Information Theory, Coding and Cryptography

Example

- A customer list might be worth Rs. 1000, an advertisement data might be worth Rs. 50,000 and the master key for a digital cash system might be worth millions.
- In the world of stock markets, the secrets have to be kept for a couple of minutes.
- In the newspaper business today's secret is tomorrow's headlines.
- The census data of a country have to be kept secret for months (if not years).
- Corporate trade secrets are interesting to rival companies and military secrets are interesting to rival militaries.
- Thus, the security requirements can be specified in these terms. For example, one may require that the key length must be such that there is a probability of 0.0001% that a hacker with the resources of Rs. 1 million could break the system in 1 year, assuming that the technology advances at a rate of 25% per annum over that period.



So, is the simple example so, may be 1000 rupee worth of customer list, or an advertisement data might be worth 50000. So, you will have to have different sizes of key for these two right. In the world stock market, secrets have to be kept for couple of minutes only before it becomes open.

And in newspaper business today's secret is tomorrow's headlines, so you only have to protect it for 24 hours. Census data of the country has to be kept secret from months. We are giving examples of different kinds of data, which need to be protected for different amounts of time, and hence a corresponding key length ok.

Corporate trade secrets have to be kept in the dark for several years, if not decades same with military secrets, diplomatic secrets right. So, for example, when we required the key length to be such that there is a probability of say 0.011 percent that hacker will be able to crack it with the resources of rupees 1 million, in 1 year. So, we can define, the security of a cryptosystem or a requirement in terms of the probability of it being hacked with a certain amount of money, and time resources being available to the eavesdropper.

(Refer Slide Time: 40:09)

Information Theory, Coding and Cryptography

Minimum Key Requirements of some operations

Type of information	Lifetime	Minimum key length
Tactical military information	Minutes/hours	56-64 bits
Product announcements	Days/weeks	64 bits
Interest rates	Days/weeks	64 bits
Trade secrets	decades	128 bits
Nuclear bomb secrets	> 50 years	512 bits
Identities of spies	> 50 years	512 bits
Personal affairs	> 60 years	> 512 bits
Diplomatic embarrassments	> 70 years	> 512 bits


So, based on this, there are some typical numbers and this really has to change depending upon the secrecy, requirements. So, write from military data, to trade secrets, to diplomatic embarrassments all of them have to be linked with keys, which are larger than certain number of bits. And these, probably have to be multiplied by a factor of 4 with the new increased processor speeds that are available today.

(Refer Slide Time: 40:47)

Information Theory, Coding and Cryptography

Message Authentication Code

- Symmetric cryptography can also be used to address the integrity and authentication requirements.
- The sender creates a summary of the message, or **message authentication code (MAC)**, encrypts it with the secret key, and sends that with the message.
- The recipient then re-creates the MAC, decrypts the MAC that was sent, and compares the two.
- If they are identical, then the message that was received must have been identical with that which was sent.




So, symmetric cryptography can also be used to address the integrate in authentication requirements, so these are requirements in real life. And the sender creates a summary of the message, or message authentication code, encrypted with the secret key, and sends it with the message. So, that recipient then re-creates the MAC, and decrypts the MAC that was sent, and compare the two. Very similar to thus hash function that we talked about.

(Refer Slide Time: 41:18)

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.



Let us quickly talk about the types of symmetric key algorithms, one is the block cipher, the other is a stream cipher. The block cipher as we discussed, breaks up the data input


into blocks typically of the size of key length, and then encodes block by block. Stream ciphers are slightly different, they are like convolution encoders, you use memory element. So, you have a memory unit, where the incoming bit stream comes in, and then you have a small algorithm, which encodes. And, so the output depends on the new input data, and what was sitting in the memory. So, very similar to the convolution encoder so, block ciphers very similar to block codes, stream ciphers very similar to convolution codes.

(Refer Slide Time: 42:08)

Information Theory, Coding and Cryptography

One Time Pad

- The one-time pad was invented by Major Joseph Mauborgne and Gilbert Vernam in 1917, and is an unconditionally secure (i.e. unbreakable) algorithm.
- The pad is a **non-repeating random string** of letters. Each letter on the pad is used once only to encrypt one corresponding plaintext character.
- After use, the pad must **never** be re-used. As long as the pad remains secure, so is the message.
- This is because a random key added to a non-random message produces completely random ciphertext, and there is absolutely no amount of analysis or computation that can alter that.



One example of one-time pad, the only known completely secure algorithm; invented in 1917's. So, the pad is a non-repeating random string of letters. Each letter on the pad is used only once to encrypt one corresponding plain text character. So, after use the pad should never be re-used. As long as the pad remains secure, the message is secure.


So, essentially you really will never know, what the true message is. This is a truly secure algorithm, because even if you decode it correctly, you have no idea, whether you have reached the correct sequence that is the beauty of one time pad. The random key added to a non-random message produces a completely random ciphertext, and there is absolutely no amount of analysis, or computation that can alter that. That is the guiding principle of a onetime pad.

(Refer Slide Time: 43:12)

Information Theory, Coding and Cryptography

One Time Pad (continued)

- If both pads are destroyed then the original message will never be recovered.
- There are two major drawbacks:
 - Firstly, it is extremely hard to generate truly random numbers, and a pad that has even a couple of non-random properties is theoretically breakable.
 - Secondly, because the pad can never be reused no matter how large it is, the length of the pad must be the same as the length of the message - fine for text, but virtually impossible for video.




If both the pads are destroyed, then the original message will never be recovered. But, there are two major drawbacks, it is very hard to generate truly random numbers right. And secondly, because the pad can never be reused, no matter how large it is, the length of the pad must be the same size as the length of the message. This is also shown also by Shannon, that in order to have virtually no mutual information between the ciphertext and the plaintext image. The key length should be as long as the message itself, and that is essentially why one time pad works, but it is practically not possible.

(Refer Slide Time: 44:00)

Information Theory, Coding and Cryptography

Steganography

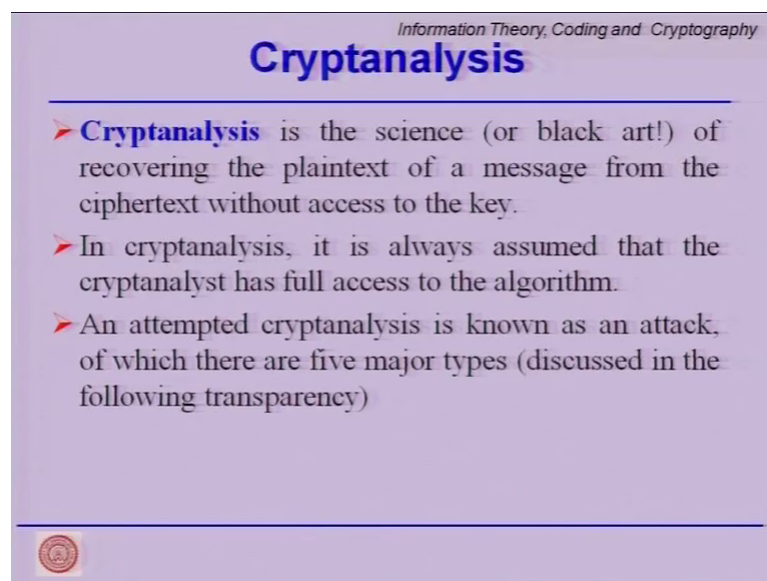
- Steganography is not actually a method of encrypting messages, but **hiding** them within something else to enable them to pass undetected.
- Traditionally this was achieved with invisible ink, microfilm or taking the first letter from each word of a message. This is now achieved by hiding the message within a graphics or sound file.
- **Example:** For a 256-greyscale image, if the least significant bit of each byte is replaced with a bit from the message then the result will be indistinguishable to the human eye. An eavesdropper will not even realise a message is being sent.
- This is not cryptography however, and although it would fool a human, a computer would be able to detect this very quickly and reproduce the original message.



A quick word about steganography, it is basically hiding the message. So, I can hide the message in a in image, I can hide the message in the least significant bits of my image. So, steganography is basically like the invisible film, invisible ink, and we can basically use different techniques to hide the message, original message in a clutter or in another message. It is not a very commonly used mechanism for cryptography ok.

So, this is an example. I take a 256-greyscale image, and I can use the least significant bit to encode my data, and this image will not be fundamentally altered, because I have just modified the least significant bits. But, there is an entire message being encoded within this image. So, but this can fool the human eye, but a computer program would be very easily able to detect and reproduce original image. So, steganography is not really in fashion.

(Refer Slide Time: 45:22)



Information Theory, Coding and Cryptography

Cryptanalysis

- **Cryptanalysis** is the science (or black art!) of recovering the plaintext of a message from the ciphertext without access to the key.
- In cryptanalysis, it is always assumed that the cryptanalyst has full access to the algorithm.
- An attempted cryptanalysis is known as an attack, of which there are five major types (discussed in the following transparency)


We will spend the last couple of slides talking about cryptanalysis because today's lecture is an overview lectures we come down to the cryptanalysis part which is nothing but a science of recovering the plaintext of a message from the ciphertext without the key that is analysis. We do how difficult it is to break into the system. So, cryptanalysis it is assume that the cryptanalyst has full access to the algorithm, we never keep the algorithm hidden. So, an attempted cryptanalysis is called an attack and the different kinds of attack.

(Refer Slide Time: 45:51)

Information Theory, Coding and Cryptography

Methods of Cryptanalysis

- *Brute force attack:* This technique requires a large amount of computing power and a large amount of time to run.
 - It consists of trying all possibilities in a logical manner until the correct one is found.
 - For the majority of encryption algorithms a brute force attack is impractical due to the large number of possibilities.
- *Ciphertext-only:* The only information the cryptanalyst has to work with is the ciphertext of various messages all encrypted with the same algorithm.
- *Known-plaintext:* In this scenario, the cryptanalyst has access not only to the ciphertext of various messages, but also the corresponding plaintext as well.



So, let us list them out. The first one is the brute force attack it is commonly used. Basically you try out all permutations at combinations right, but for majority of the encryption algorithm, the brute force attack is simply impractical due to the large number possibilities, so that attackers resort to other techniques. For example, ciphertext-only attack where the only the information of the cryptanalysis has to work with the ciphertext of various messages all encrypted with the same information.


So, it is slightly more difficult, but you know the body of ciphertext available to the cryptanalyst. All we can have a known-plaintext attack in which the cryptanalysis has an access to not only the ciphertext of various messages, but also the corresponding plaintext. So, he or she can compare the plaintext with the ciphertext and try to derive some edition information.

(Refer Slide Time: 46:49)

Information Theory, Coding and Cryptography

Methods of Cryptanalysis (cont.)

- *Chosen-plaintext:* The cryptanalyst has access to the same information as in a known-plaintext attack, but this time may choose the plaintext that gets encrypted.
 - This attack is more powerful, as specific plaintext blocks can be chosen that may yield more information about the key.
 - An adaptive-chosen-plaintext attack is merely one where the cryptanalyst may repeatedly encrypt plaintext, thereby modifying the input based on the results of a previous encryption.
- *Chosen-ciphertext:* The cryptanalyst uses a technique called differential cryptanalysis, which is an interactive and iterative process.
 - It works through many rounds using the results from previous rounds, until the key is identified.
 - The cryptanalyst repeatedly chooses ciphertext to be decrypted, and has access to the resulting plaintext. From this they try to deduce the key.



We can make things easier for the cryptanalyst by giving him the option of a chosen-plaintext attack, where the cryptanalysis has the access to the same information as in a known-plaintext attack, but this time may choose the plaintext that get encrypted. So, he or she may first try to look at a chain of all a's, and see how the encoder block turns out and then all b's and then all c's.


So, basically you can choose and design the attack by putting a chosen-plaintext attack then you can have a chosen-ciphertext attack where the cryptanalysis uses a technique call the differential cryptanalysis which is an interactive and iterative process. And it works by several round and you keep on refining the ciphertext chosen ciphertext so as to guess how the encoding operation works. So, this is a different levels and the freedom we want to grant to the cryptanalyst to see how strong the encoding algorithm is.

(Refer Slide Time: 47:59)

Information Theory, Coding and Cryptography

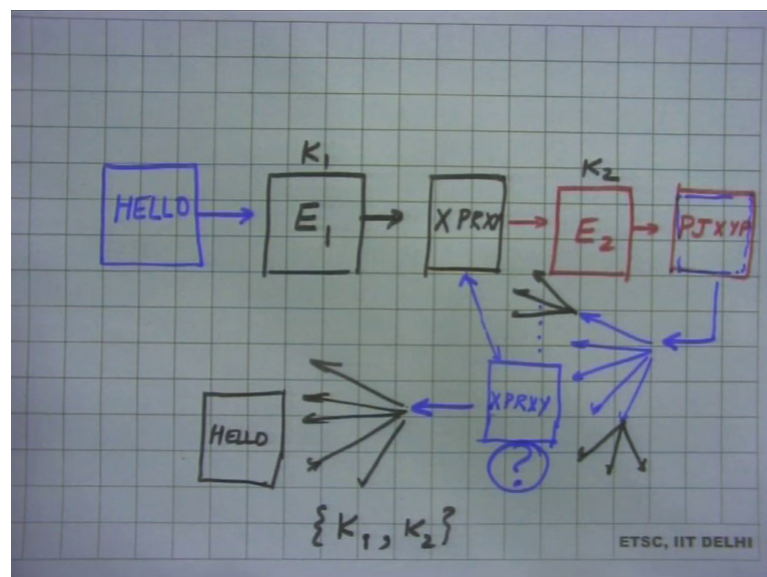
Brute Force Attack

- Regarding brute force attack, there are a couple of pertinent questions:
- What if the original plaintext is itself a cipher? In that case, how will the hacker know if he has found the right key.
- In addition, is the cryptanalyst sitting at the computer and watching the result of each key that is being tested?
- Thus, we can assume that brute force attack is 'impossible' provided long enough keys are being used.



So, brute force attack of course, there are some questions regarding it what is the original plaintext is a ciphertext. What if you have run two ciphers one after the other. So, even if you get a decoded message, you will not be able to see what was original message. Let us look at it graphically.

(Refer Slide Time: 48:31)



So, suppose you have a plain text image say hello. And you run it through an encryption algorithm let say E_1 , and you get an output and then you run it through one more round. And you get finally, now the eavesdropper the adversary runs a brute force attack. So, it

works with the ciphertext and mounts a brute force attack. And so he will have several options if the attack is exhaustive out of the many outputs at the attack generates one of them will be actually the correct one, but this itself has been encoded. So, the attacker does not know whether he is stumble upon the right answer or whether that key was the right key because this itself does not make sense.

So, the person has to not stop here, but go and go ahead and carry out the second attack. And look at all possibilities similarly for all them right that is go around the second round and in this may be one of them will generate the hello. So, the eavesdropper essentially is working with now the larger size of the key if this 1 was K_1 and K_2 . So, it is looking at K_1 , K_2 as a pair right. So, you have just made is life more difficult that is about it. This is the first point that we observe from the brute force attack method.


Second question is who is sitting at the computer and watching all the output. Can the computer be smart enough to look at and raise a flag, may be it can help us reduce the such space right. So, typically we say that if the keys are long enough 128 bit is not long enough, but 1024 bits is definitely long enough. So, you can almost say that the brute force is not really going to work and the cryptanalyst will try to resort to other techniques.

(Refer Slide Time: 51:34)

Information Theory, Coding and Cryptography

Other Techniques

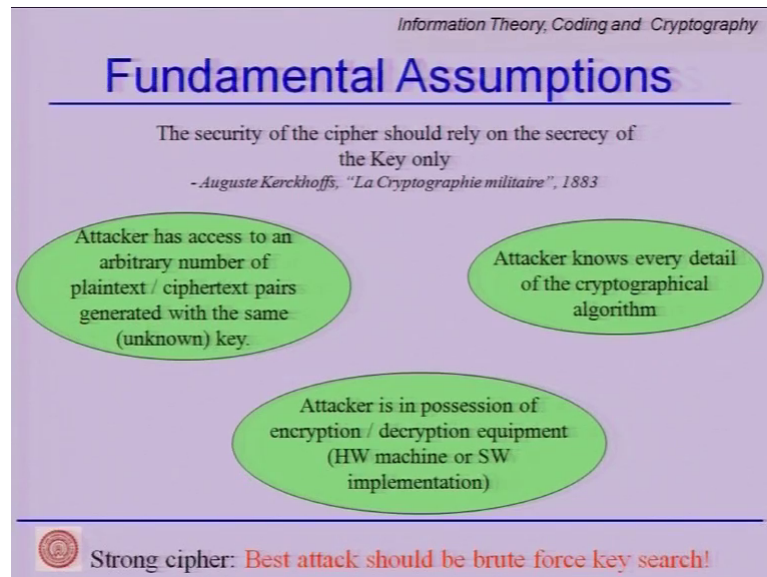
- **Differential cryptanalysis:** This technique uses an iterative process to evaluate cipher that has been generated using an iterative block algorithm (e.g. DES). Related plaintext is encrypted using the same key.
 - The difference is analyzed.
 - This technique proved successful against DES and some hash functions.
- **Linear Cryptanalysis:** In this, pairs of plaintext and ciphertext are analyzed and a linear approximation technique is used to determine the behaviour of the block cipher.
 - This technique was also used successfully against DES.
- **Algebraic attack:** This technique exploits mathematical structure in block ciphers.
 - If the structure exists, a single encryption with one key might produce the same result as a double encryption with two different keys.
 - Thus the search time can be reduced.



So, the other techniques will quickly mention are differential cryptanalysis, linear cryptanalysis and algebraic attacks will not really go into the detail because this is just an

introductory lecture, but in differential cryptanalysis this technique uses an iterative process to evaluate the cipher text that has been generated using an iterative block algorithm right; similarly, linear cryptanalysis and algebraic attacks.

(Refer Slide Time: 51:58)




So, we just make sure that we believe the following. The attacker has an access to an arbitrary number of plaintext and ciphertext pairs ok. Attacker knows every detail of the algorithm. And attacker is in possession of the encryption decryption equipment if we are doing hardware encryption. We give them all of this and then we say go ahead and find out.

(Refer Slide Time: 52:26)

Information Theory, Coding and Cryptography

Politics Of Cryptography

- Widespread use of cryptosystems is something most governments are not particularly happy about – precisely because it threatens to give more privacy to the individual, including criminals.
- For many years, police forces have been able to tap phone lines and intercept mail, however, in an encrypted future that may become impossible.
- This has led to some strange decisions on the part of governments, particularly the United States government.
- In the United States, cryptography is classed as a munition and the export of programs containing cryptosystems is tightly controlled.




Well, just slide on politics of cryptography because this is very important. This pertains to the national security for most countries. And then there is a lot of export control put on cryptographic algorithms which are used by different countries. And we all know that phone lines are tap and then people use encryptions on the phone lines. And so there is a whole range of politics attached to cryptography and cryptographic research. So, it was important to bring of this point.

(Refer Slide Time: 53:04)

Information Theory, Coding and Cryptography

Summary

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



Indian Institute of Technology,
Delhi

41

Ranjan Bose
Department of Electrical Engineering

With that we come to the end of this lecture what we have covered so far is a basic introduction to cryptography. We talked about the key symmetric key encryption, asymmetric key encryption, and we spend some time on cryptanalysis. With that we come to the end of this lecture.