Hardware Security

Prof. Debdeep Mukhopadhyay
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 08
Background on Cryptography, Cryptanalysis and Advance Encryption Standard (AES)

Welcome back. So, today we shall we studying on our next topic which is essentially a Background on Cryptography and also to have an understanding of what is meant by Cryptanalysis. And finally, we shall be talking about the Advanced Encryption Standard or AES.

(Refer Slide Time: 00:36)

CONCEPTS COVERED

Concepts Covered:

❑Goals of Cryptography

❑Types of Cryptography

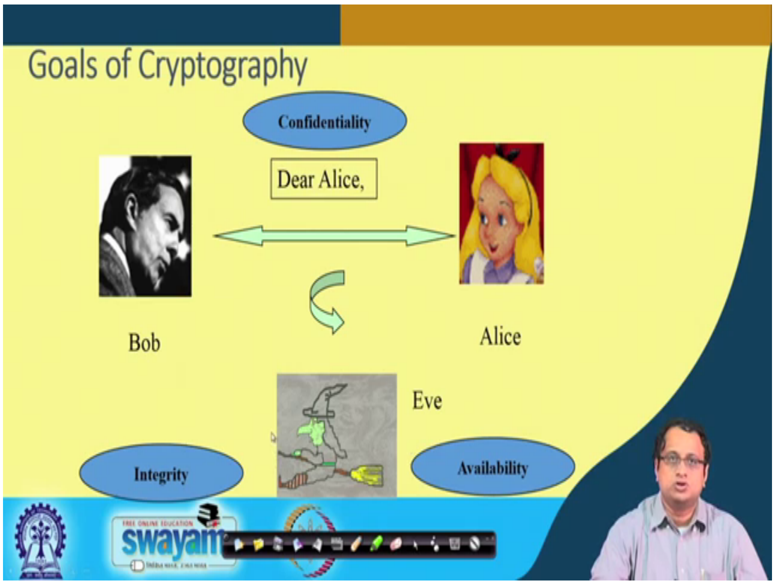
❑Block Ciphers

❑AES: Advanced Encryption Standard

So, to start with our concepts that the concept that we shall be covering in today's class; we shall be trying to understand the goals of cryptography, try to understand the various types of cryptographic algorithms; I would say like the broad classifications of cryptographic algorithms. In particular, we shall be trying to look into block ciphers which is a very common form of cryptographic algorithm which is used to encrypt bulk volumes of information or data.

And we shall be looking into one particular encryption or block cipher which is called as a AES or Advanced Encryption Standard; which is the de facto worldwide standard for block ciphers.

(Refer Slide Time: 01:15)



Goals of Cryptography

So, to begin with let us try to look into the goals of cryptography. So, the goals of cryptography can be are typically depicted by these three party scenario; where you have got two legitimate players Bob and Alice who are essentially trying to share some information over an untrusted channel. So, this untrusted channel is essentially also observed by an attacker or an adversary which is typically denoted as Eve.

So, Eve is interested to know; what is the communication which is essentially being done by between Bob and Alice? So, in order to prohibit Eve from getting access to this information; Bob tries to resort to cryptography. What Bob tries to apply cryptography and tries to apply the in the data or tries to apply it on the data and communicate this information to Alice.

So, there can be different goals of cryptography; confidentiality is one of the most important goals as I said, because Eve will be probably curious to know what is the information being exchanged and in that context Bob of an encrypts this information and sends it to Alice. So, once it is encrypted the idea is that it looks like a jumbled piece of data from which Eve is not able to understand or make any inference about the actual content of the message.

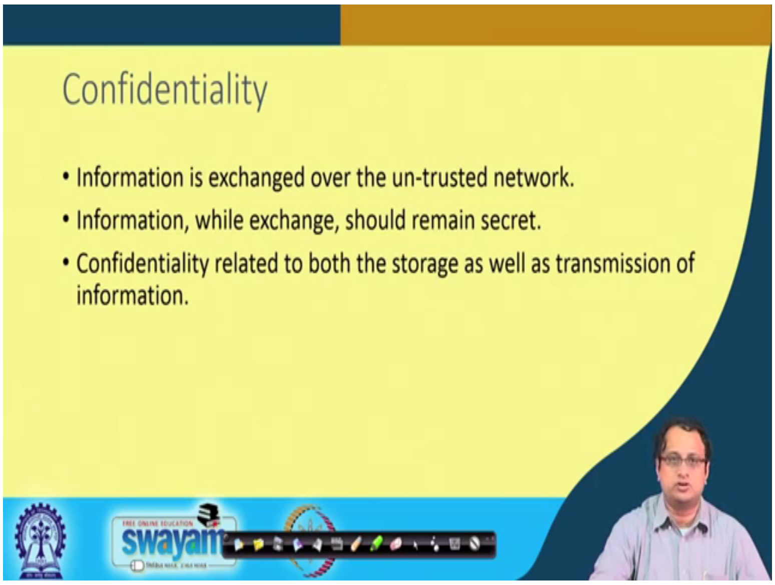
So, now I would like to mention here that although we often you know like kind of synonymously use cryptography with confidentiality, but that is not the sole goal of encryption or cryptography that is not the sole goal of cryptography. So, there may be other important goals for example, I may be only bothered about integrity of information.

So, I may be only bothered to ensure that I do not really; I am not really bothered to ensure that Eve does not have access to information, but rather I may be bothered to know to ensure that Eve is not able to corrupt that information or he is not able to modify that information. So, therefore right while confidentiality is often insured by encryption; for integrity there may be other primitives which are used. So, it could be typically a hash function or a message authentication code which could be deployed to achieve integrity.

But at the end of the day you know like with several goals like you may have confidentiality as your goal, integrity as your goal and you may try to apply cryptographic algorithms. But all cryptographic algorithms comes with over it; comes with essentially you know like making the process flow, kind of making the process cumbersome. So therefore, right we should ensure at the end of the day that data should be available.

So, we should ensure the data should be available to the; to the legitimate persons like Bob and an Alice and Eve should not be able to somewhat availability of information ok. So, there may be there should be suitable mechanisms which also allows you or which allows you to have; I would say like you know like uninterrupted access to information or on and inter to access to data.

(Refer Slide Time: 04:06)



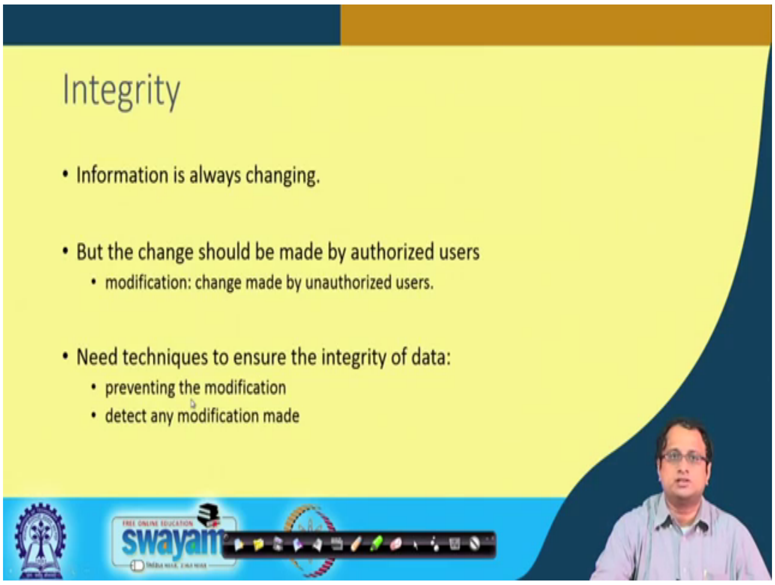
Confidentiality

- Information is exchanged over the un-trusted network.
- Information, while exchange, should remain secret.
- Confidentiality related to both the storage as well as transmission of information.

So therefore, right I mean let us try to take a; you know like one by one look into these various objectives; so the first objective whether I said is confidentiality. So, information is exchanged over the untrusted network. So, information while exchange should remain secret. That means, you know like when you are exchanging information over the untrusted network, you should ensure that it should remain secret to the legitimate players; to the sender and to the receiver; that means, Alice and Bob in our context.

So, confidentiality is related to both not you know like not; I mean not only with respect to transmission of information, but also it could be with respect to data addressed; that means, it should be also it could be it should be related to also data; which is kept in your storages. So, it could be in your disks it could be in your drives and other storage media.

(Refer Slide Time: 04:51)



Integrity

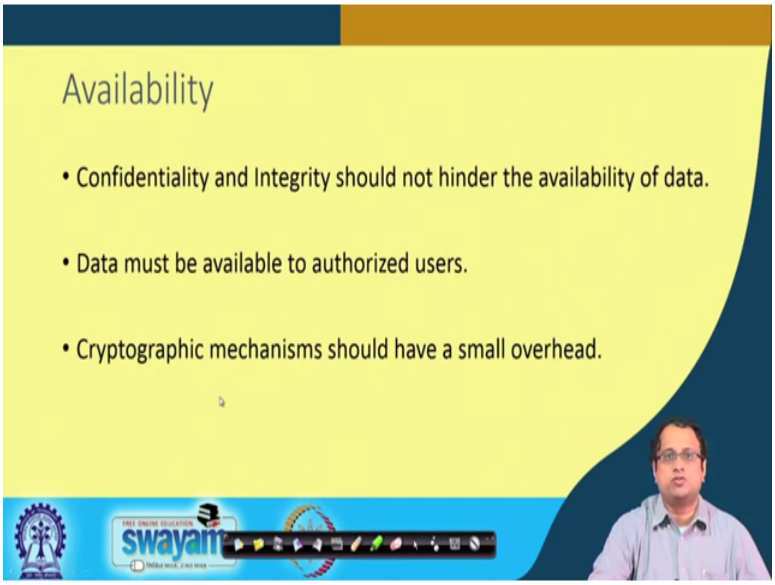
- Information is always changing.
- But the change should be made by authorized users
  - modification: change made by unauthorized users.
- Need techniques to ensure the integrity of data:
  - preventing the modification
  - detect any modification made

So likewise right I said that integrity is also an important objective. So, as we know that information right one of the important criteria of information is that it is always changing. And that is probably even more relevant with today's talk discussions on you know like big data and so on, where information is continuously being generated and is an expensive data base right.

So, but the charge; so therefore, right we should ensure that the responsibility the; of the of the persons who are you know like essentially providing service should be that unauthorized people should not be able to modify that information. So that means, you know like for example, if I have got some amount of data which is for example, measuring the amount of money that I have in my bank and unauthorized person should not be able to modify that data ok.

Because you can easily understand that if that data is modified then that can that can lead to catastrophic consequences. So, therefore we need to techniques to ensure that there is integrity of data which means like somebody should be; who is not licensed or is not legitimate should not be able to modify the data. And moreover like if there is any modification which is going down; I should be able to readily understand that there has been a modification being done. That means, it should be detected any modifications should be easily detected and should be essentially you know lead to follow of measures to ensure that there is in really integrity of information.

(Refer Slide Time: 06:20)



Availability

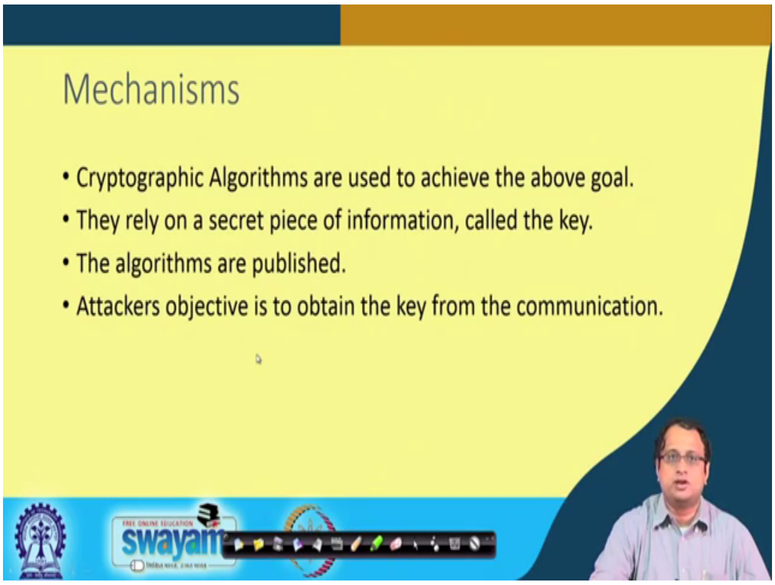
- Confidentiality and Integrity should not hinder the availability of data.
- Data must be available to authorized users.
- Cryptographic mechanisms should have a small overhead.

So, likewise confidentiality and integrity should not hinder the availability of data. That means I mean I may apply cumbersome techniques, but at the same time right. We should ensure that availability is a very important goal right I want to make things efficient.

So, therefore, right data must be available to authorized users at any time. And therefore, we should also always keep in mind cryptographic mechanisms, which we are adopting should essentially be implemented in a manner that they essentially incurred less overhead; in terms of various performance foot prints it should have less food it should have a less footprint ok. And that essentially makes hardware design so important because when you have hardware design for cryptographic algorithms, you essentially have an opportunity of making them optimized ok.

But at the same time right we also need to tackle other criteria like for example, various kinds of threats which can also be targeted specifically for hardware ok. But in terms of performance right; hardware provides immense opportunity of realizing lesser footprints on various criteria ok. So, for example, like you can as I said probably in one of my introductory classes that if you go for hardware design of complex cryptographic algorithms, there is this there is an there is an order improvement over your software designs. And therefore, we need to probably study about how we can make efficient hardware implementations of cryptographic algorithms.

(Refer Slide Time: 07:47)



Mechanisms

- Cryptographic Algorithms are used to achieve the above goal.
- They rely on a secret piece of information, called the key.
- The algorithms are published.
- Attackers objective is to obtain the key from the communication.

So, therefore, right you can actually definitely have different mechanisms like cryptography off cryptographic algorithms, you can have encryption, you can have hash functions, you can have signature schemes; you can you know you can apply public key cryptography for signatures. So, there you know like lot of cryptographic algorithms which you have; which have been developed, but we need to understand about where to apply what and that is typically what we studying in cryptographic course.

So therefore, one of the like I would say salient features of all these cryptographic algorithms is that the algorithms are published. That means, the algorithms that we deploy for encryption or for signatures; they are all published. That means, they are available to even our adversaries; what is not available to an adversary is this piece of secret which is called often as the key ok. And therefore, right the objective of the attacker is to obtain this key from the communication.

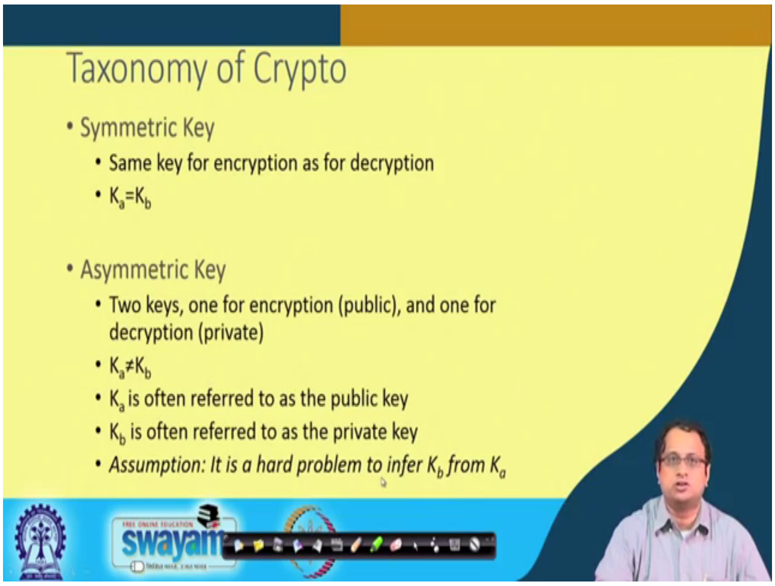
And the objective of the defender or the designer is to ensure that the information that is exchanged; it could be the cipher text, it could be the plaintext, it could be the cipher text plaintext combinations that should not reveal information about the secret key and therefore, what we write as designers want to protect is the secret key ok. Therefore, right if you if you try to understand in a more formal setting what is a cryptosystem?.

So, essentially cryptosystem is a five tuple often denoted by this like P, C, K, E and D where P stands for the set of possible or finite set of possible plain texts, C stands for the finite set of possible cipher texts, K is the key space ok; it is a finite set of possible keys from which I choose a specific key. Now, the moment I choose a specific key for example, a small k for example, then two things this gets decided; one is what is called as an encryption rule and the other one is what is called as a decryption rule.

So, the encryption rule which is denoted as say e k belongs to this encryption family and likewise d k belongs to this decryption family. So, that is a very important criteria which you can easily probably understand is that the process should be reversible right; that means, when I encrypt a piece of information and when I decrypt it; I should get back the original data. And that is being denoted here in a more formal way which says that for each e k; that means, which is essentially a mapping from the plaintext space to the ciphertext space and for each d k which is essentially the divorce mapping from ciphertext to the plaintext space.

So, these are functions such that for all x which is essentially belonging to the plaintext when I apply the encryption function on x and I get e k of x and then I apply d k of x; which is the decryption function, I should get back x. So therefore, this process should be reversible; so this invertibility is an is a feature which all encryption whether it is you know like symmetric key or public key as will soon study; all of them should satisfy this property there is a salient feature because we want to decrypt back the original data.

(Refer Slide Time: 10:44)



Taxonomy of Crypto

- Symmetric Key
  - Same key for encryption as for decryption
  - $K_a=K_b$
- Asymmetric Key
  - Two keys, one for encryption (public), and one for decryption (private)
  - $K_a \neq K_b$
  - K_a is often referred to as the public key
  - K_b is often referred to as the private key
  - *Assumption: It is a hard problem to infer K_b from K_a*

So, therefore right there are two different broad classes of cryptography which people have studied. The first one is probably very intuitive which is called a symmetric key cryptography; which means essentially it means those encryption algorithms, where the encryption key and the decryption key are same. That means, if I want to encrypt a piece of information and if I want to decrypt the cipher; then I essentially share that key with my recipient ok.

So, there has to be some kind of cross state channel through which that secret key has been shared. Now you can immediately; immediately understand that if there are n players in a network and if all of them are connected with each other then the number of keys can increase exorbitantly; can increase quadratically. And therefore, we need to have efficient mechanisms to do this key exchange and therefore, right there is another class of tiptographic algorithms which are called as a symmetric key cryptography which essentially gives a very nice solution for this particular problem.

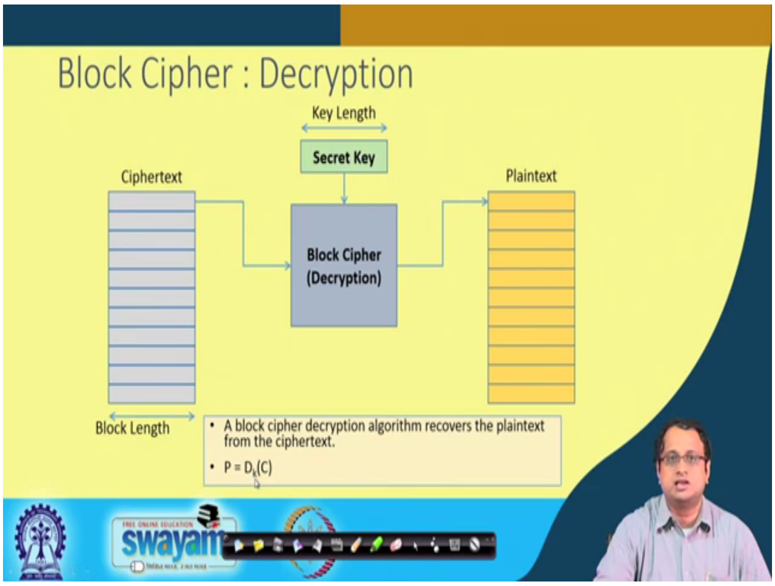
So, therefore, in this setting of asymmetric key cryptography; there are two keys. One which is called as a public key the other one which is called as a private key or the decryption key; the idea is that the first important things to note is that the encryption key and the decryption key or the public key and the private key are not the same ok.

So, what we often do is that with public key we can encrypt; that means, suppose I want suppose Alice wants to send a piece of information to Bob. So, therefore, what Alice does is that Alice encrypts it and it encrypts it with bobs public key. And now Bob since Bob only has his private key can only decrypt that message; that means, anybody can encrypt that message and send it to Alice, but I mean send it to bob, but only Bob can decrypt it because its bobs own private key ok.

So therefore, right I mean that essentially is nice because in that case the key exchange problem is nicely handled in that case. And the assumption behind these class of problems is often rely based upon a candidate hard problem; that means, a problem which is believed to be hard. The idea is that it is believed to be hard to derive the private key from the public key; note that Alice’s public key is known to everyone.

But what is important is to ensure that it should be difficult for Eve; who is an attacker to infer the corresponding private key of Alice from the public data and that the way that this kind of reasonings are done is often based upon a hard problem. So, the hard problem could be for example, factorization or some other hard problem which is believed to be hard; which is believed to be computationally intensive.

(Refer Slide Time: 13:18)



Block Cipher : Decryption

- A block cipher decryption algorithm recovers the plaintext from the ciphertext.
- $P = D_k(C)$

Therefore, let us first of all try to look into the one in very important class of symmetric key encryption which is called as block cipher and block ciphers are typically used for bulk data encryption. So, in therefore in block cipher what essentially is done is that as the name suggests right; it basically encrypts in bits of plaintext which is called as a block. So, what essentially is done here is that you take the plaintext and you divide it into chunks of data; all of them are like blocks of information and then you start applying your block cipher.

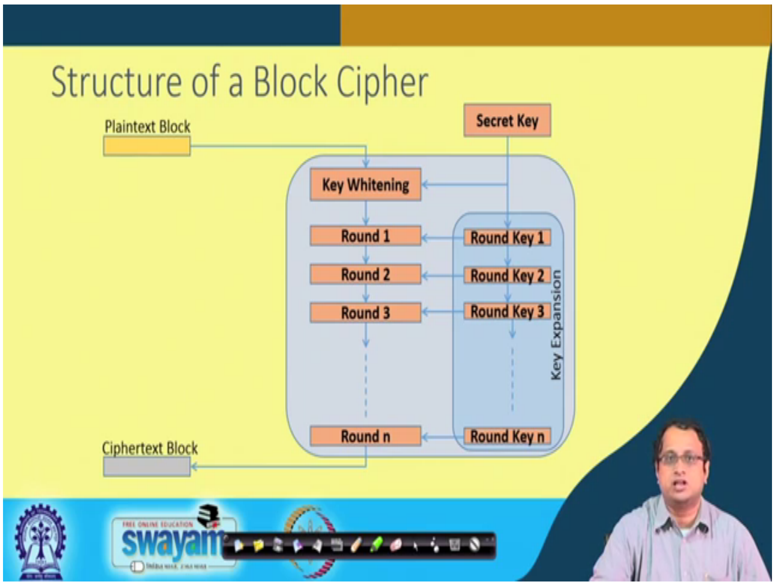
So, the block cipher takes one block of data, takes the secret key and then obtains another block of cipher text. This process could be either repeated block by block or there could be some modes where you do chaining for example, ok. But the whole idea is that when you are having this kind of encryption algorithm; then this encryption algorithm applies on one block of data. Say for example, it could be 128 bits of data as we will see in the case of AES 128 and produces hypertext; which is also 128 bits of data.

Now the key right is often also of the similar dimension; so it could be like say 128 bits or say 256 bits on which you are applying your encryption process. So therefore, the block cipher takes two important two inputs ok; it takes the plaintext and the key and produces a cipher text. So, it basically it is a mapping from the plaintext and the key to a cipher text space ok. So therefore, that is denoted here as this function; which is C is equal to E k P which means like it is an encryption function which is kept kind of parameterized by the key and it operates on a plaintext P and produces a cipher text C ok. So, now you can if you want to decrypt it as you can easily understand then you need to have an invertible process.

So, you did not have a reversible process right; so therefore, what the decryption function does? It applies on the cipher text again on a block of data, again apply the same secret key; remember it is a symmetric key algorithm. So, it has got the same secret key on which this also works. And therefore finally, produces that the starting the block the starting block of the plaintext ok. And that by that way the plaintext is recovered block by block. Again you can if you are depending upon the mode you in which you are applying the encryption; you have to apply a suitable mode for decryption.

So, therefore the block cipher decryption algorithm is denoted by this function where I take the C which is my ciphertext. And again I take my function which is denoted as D k and I obtain back the plaintext ok.

(Refer Slide Time: 15:44)



Structure of a Block Cipher

Therefore right if I want to know zoom into the block cipher for example, then this is how the internal of a block cipher typically looks like. So, these block ciphers are water often called as product ciphers or iterative ciphers.

So, what it does here is that it takes a plaintext block; it takes a secret key and the first step is usually a key whitening step. This key whitening is usually applied or obtained by some kind of reversible key mixing process; a very common form of reversible key mixing is by applying a bitwise exclusive or so therefore, I take this plaintext, I take the secret key and I do an exorb between the plaintext and the key and obtain the input to the first round.

And then I have a period you know like a very very regular structure of applying the rounds; that means, rounds essentially are constituted by some kind of operations; these operations are applied one by one in an iterative manner. For example, in this case I have shown that there are n rounds. This number of rounds is often fixed by something which is called as script analysis which is essentially an evaluation methodology of the cipher against various known attacks. And then once you know that the n rounds are required by the cipher; then for every round you also need to have something which is called as a round key.

So, this round key is derived by an algorithm which is called as the key expansion algorithm or also called as the key scheduling algorithm. So, this key scheduling algorithm takes the secret key and derives the round keys one by one. So, here you can see that from the secret key we have got the round key 1, round key 2 and so on till round key n and all of these round keys are mixed with these rounds ok.

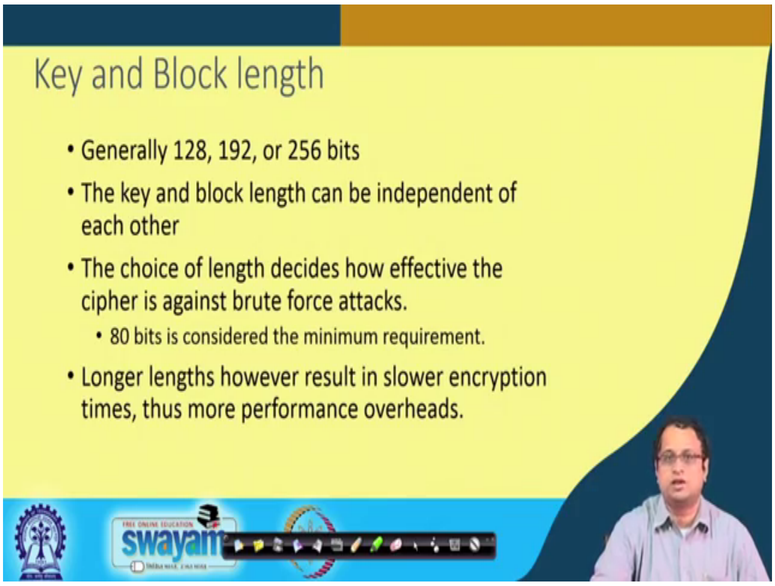
So, this mixing is also done by a similar step as the key whitening step for example, it could be done through a bitwise exclusive or ok. So, you can also apply other kind of whitening things for whitening steps for example, you can probably try with an interior module or addition also. So, there could be other kind of reversible functions which you can also try to apply over here, but the most common is probably exclusive what where you do an XOR between the plaintext and the secret key or the key components. So, finally, after n rounds you obtain an output which is called as a cipher ok.

Now, you can you should note here at this point that what the designer wants you to essentially see is that it once that the adversary should only see these plain text and the cipher text block. It wants that the cipher that the adversary should not see any intermediate data. If the adversary is able to see any intermediate data then the designer cannot give any guarantees of security ok. And what we will see later on in side channel attacks which we try to do in various hardware designs on hardware various hardware designs is that through side channels; we get try to get information about these internal rounds. And therefore, we try to retrieve the key in very in a very efficient manner ok.

But in a very black box sense what I want to show here is that that attacker should only see the plaintext and the cipher text. And then the mathematics or the algorithm guarantees that there should not be any efficient attacked better than the brute force search. That means, for example, if the key is of dimension 128 bit; then a brute force search would be a 2 to the power of 128 guess. That means, I guess with the complexity of 2 to the power of 128.

So, what I want as a designer is that there should be no attack which is better than this brute force search.

(Refer Slide Time: 19:04)



Key and Block length

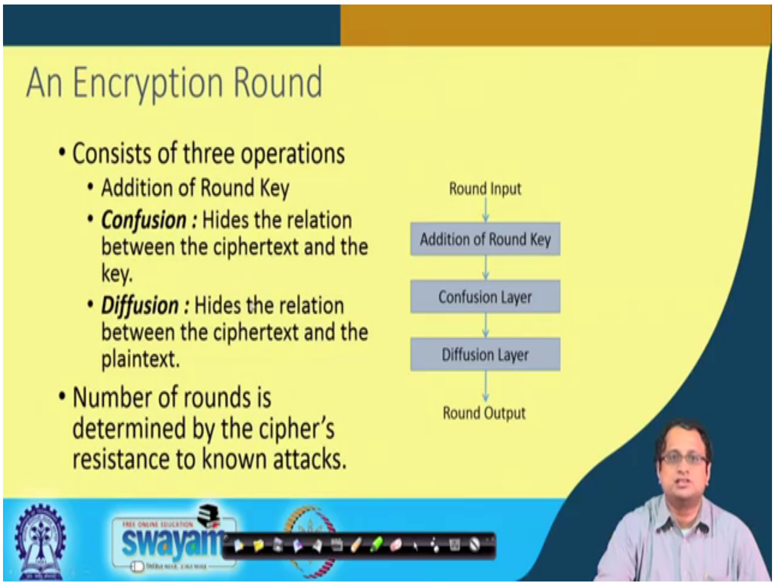
- Generally 128, 192, or 256 bits
- The key and block length can be independent of each other
- The choice of length decides how effective the cipher is against brute force attacks.
  - 80 bits is considered the minimum requirement.
- Longer lengths however result in slower encryption times, thus more performance overheads.

So, therefore, right typically the key lengths and the block lengths are preferred as 128 bits, 192 or 256; the key and the block length can be independent of each other. So, therefore, you can have for example, our cipher where there is 128 bit blocks and there the key I mean the key right can be independently chosen from the block length that it can essentially make it either 128 or 192 or 256 ok. You can also check change the block length to say 192 and again you can make independent choices of the key length.

So, therefore, right the key length and the block length could be independently chosen of each other. And the choice of the length often depends upon how effective the cipher is against brute force search ok. So, therefore, typically it is assumed that we should have at least 80 bits of security to ensure that we have got you know like enough amount of protection against computationally bounded adversaries and the longer length should you know like we can of course, keep longer length.

But you should also again keep in mind that if you have longer length then that will incur into more performance overhead. So, you will if you take more time; it will be you know like if you are talk talking about hardware design, it will consume more power, it will have more area requirement or resource requirement. So, we should try to you know like provide good amount of security with minimum resource that is always an objective of both algorithm designers or both cipher designers; as well as people who are implementing them who wants to make efficient implementations out of ciphers.

(Refer Slide Time: 20:27)



An Encryption Round

- Consists of three operations
  - Addition of Round Key
  - ***Confusion :*** Hides the relation between the ciphertext and the key.
  - ***Diffusion :*** Hides the relation between the ciphertext and the plaintext.
- Number of rounds is determined by the cipher's resistance to known attacks.

Round Input

Addition of Round Key

Confusion Layer

Diffusion Layer

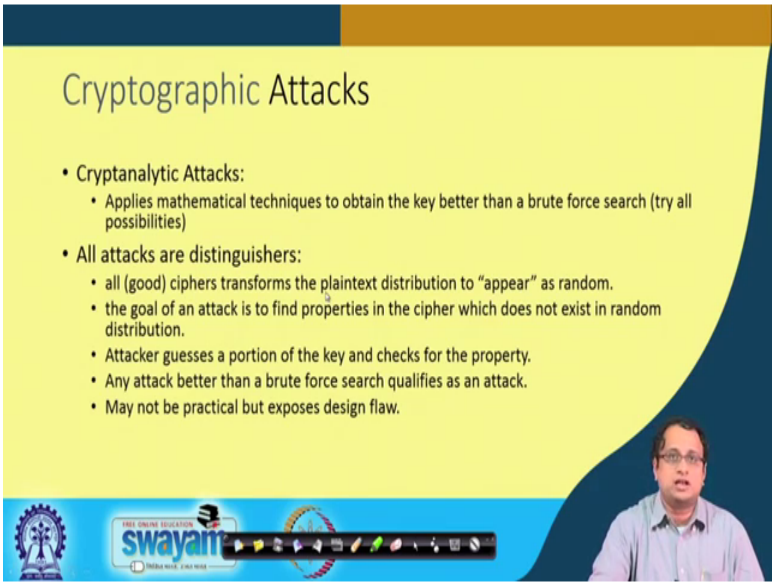
Round Output

So, therefore right, if just want to also look into what are the important criteria inside an encryption round. So, again as we have seen that in block cipher is made of rounds, but if you go into a round right; you will typically see that the cried that there are two important blocks in a round. So, one of them essentially tries to hide the relationship between the cipher text and the key and that is called as confusion. And the other one is what tries to hide relationship between the cipher text and the plaintext and that is what is called as diffusion.

So, therefore right you will see that in a block cipher round typically there are the constructions; which provides both confusion and diffusion. So, typically the artifact or the block which produces or provides confusion is called as S box or substitution box. So, we will see in more details as we study AES and other ciphers and the other layer which is called as diffusion is typically produced provided by permutations by you know like linear permutation; so linear mappings.

And then as I said the third important block or three that the third important component of the round is where you are mixing the round key with your state ok. And that is typically done by a round key process which is often realized by exclusive hours. So, the number of rounds is determined by the ciphers resistant to known attacks and that is called as script analysis ok.

(Refer Slide Time: 21:46)



Cryptographic Attacks

- Cryptanalytic Attacks:
  - Applies mathematical techniques to obtain the key better than a brute force search (try all possibilities)
- All attacks are distinguishers:
  - all (good) ciphers transforms the plaintext distribution to "appear" as random.
  - the goal of an attack is to find properties in the cipher which does not exist in random distribution.
  - Attacker guesses a portion of the key and checks for the property.
  - Any attack better than a brute force search qualifies as an attack.
  - May not be practical but exposes design flaw.

So, therefore, right talking about cryptographic attacks; so what essentially you know like. So, therefore, the overall subject is what is called as cryptology and there are two important components. One is where you talk about cryptography, where you design things and the other one is what is called as script analysis; where you try to analyze your ciphers or you try to attack your ciphers, but with the objective of analyzing them.

So, what we try to do in cryptanalytic attacks is we try to apply mathematical techniques to obtain the key better than a brute force search. Because a brute force search would mean that suppose there is an n bit key then that would imply a 2 to the power of n guess.

And we try to ensure that our construction should not be being it should not be able to be attacked by an adversary who has access to plaintext cipher text plaintext cipher text combination; whether you know like when it is knowing the plaintext or whether it is able to choose the plaintext should not be able to do an attack which is better than a brute force attack or brute force attack.

So, therefore right we will see that most of the attacks that we essentially see essentially are basically distinguish or based on something which is called as a distinguisher. Because a good ciphered essentially tries to randomize things ok; for example, right he tries to transform the plaintext distribution to appear as random in the cipher. So, the goal of an attack in general is to find out properties in the cipher which does not exist in random distribution.

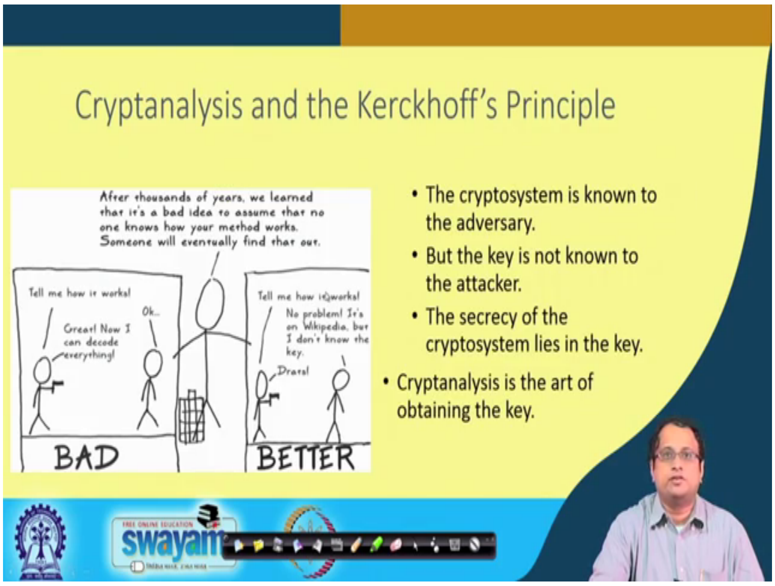
So, you may probably appreciate by this fact that suppose you have got a plaintext size of say 128 bits and a key size of 128 bits; it is not possible for someone write to evaluate all the possible mappings because simply because the mapping is huge. So, therefore the attacker always tries to you know like tries to see or tries to kind of explored, whether there is some property which the cipher still has which makes it distinctly different from a random distribution ok.

And that is really a challenging objective from the designers perspective because the designer cannot simply see all the possible; all the possible configurations because the space is huge. So therefore, right what the attacker tries to do is that in their in efficient attacks is tries to do something which is like a divide and conquer attack. So, what it tries to do is that it tries to guess a portion of the key and then it checks for the property.

So, any attack any attack which is better than a brute force search will qualify as an attack. So, therefore, right if the attacker is able to find out a property which essentially is distinctly different from a random distribution then; that means, that that key essentially or that portion of the key gets explored. And if you can do it in a divide and conquer manner then you essentially have an efficient key retrieval process which is better than a brute force search.

So, the objective of the designer is to ensure that our designs are not amenable to such a brute a better than brute force such attack and therefore, they are resistant against such attacks ok. And may not be you know like often these attacks may not be practical; that means, suppose there is in a rather than 2 to the power of 128; if I do an attack we say complexity of 2 to the power of 120; it is not a practical attack but you definitely expose a design flaw which we would like to prevent.

(Refer Slide Time: 24:49)



Cryptanalysis and the Kerckhoff's Principle

After thousands of years, we learned that it's a bad idea to assume that no one knows how your method works. Someone will eventually find that out.
Tell me how it works!
Ok...
Great! Now I can decode everything!
BAD
Tell me how it works!
No problem! It's on Wikipedia, but I don't know the key.
Drats!
BETTER

- The cryptosystem is known to the adversary.
- But the key is not known to the attacker.
- The secrecy of the cryptosystem lies in the key.
- Cryptanalysis is the art of obtaining the key.

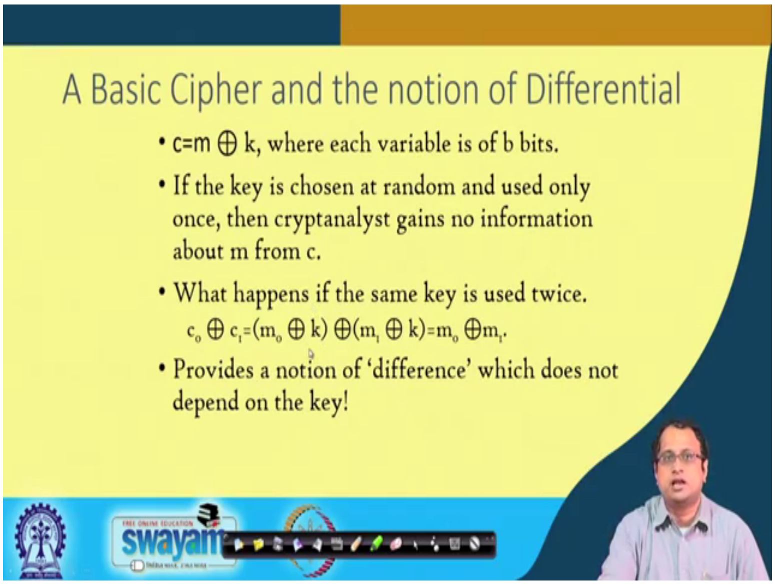
So, therefore, right I mean I mean one very important message is that there is a principle which is called as Kerckhoff's principle which says that you know like the cryptographic algorithm is always assumed to be known to the adversary. And this is something that we have understood historically that we may try to develop proprietary algorithms, but they are often you know like various ways through which the proprietary algorithm eventually comes to the public ok.

So, therefore people can either reverse engineer or even they can compromise a person who has knowledge about the about the algorithm. So, rather a better approach is that we will try to you know like rely on something which is called as a scheme ok. We will try to rely upon the fact that there is a piece of information which is called as the key and the attacker does not know the key.

And therefore, the security should be reliant on the key and therefore, even if you know like get hold about the encryption algorithm; then the security is not compromised.

Moreover, it also tries to tell that you know like you; you essentially you know like the moment you open up your algorithm to public scrutiny, then your algorithm gets better analyzed right. And therefore, you really come to know about some kind of flaws which may be you know like we may which may be have been overlooked by the designer.

(Refer Slide Time: 26:03)



A Basic Cipher and the notion of Differential

- $c=m \oplus k$, where each variable is of b bits.
- If the key is chosen at random and used only once, then cryptanalyst gains no information about m from c.
- What happens if the same key is used twice.
  $c_0 \oplus c_1=(m_0 \oplus k) \oplus (m_1 \oplus k)=m_0 \oplus m_1$.
- Provides a notion of 'difference' which does not depend on the key!

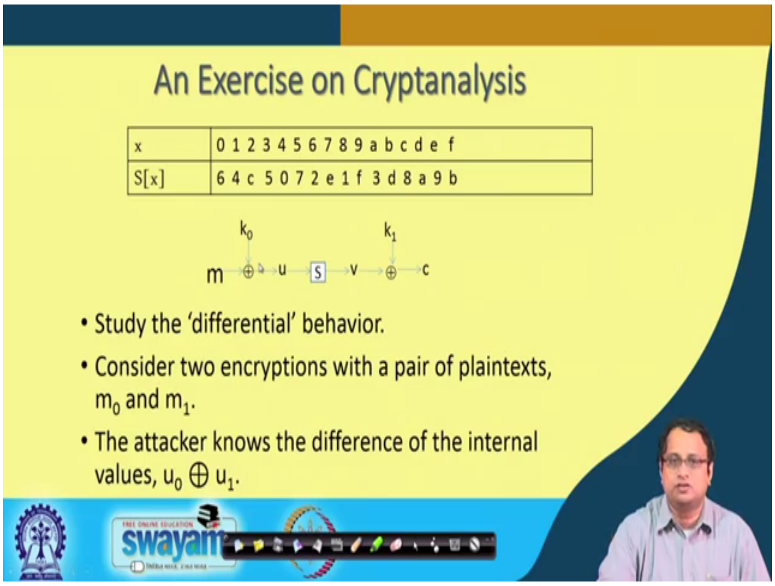
So therefore, right I mean there are different ways in which you can do crypt analysis. So, I will try to give you a basic notion about an attack an attack strategy which is called as differential analysis. So, therefore, right when you talk about differential analysis then the usual; so as I said that suppose you consider a cipher which is made of c equal to m XOR k where m is the message and k is the key ok.

So, therefore, write each variable you can assume to be b bit of information. So, if the key is chosen at random and used only once; then the crypt analysis is guessed no information about m from c what happens right if the key is used twice. So, you will see that you know like. So, this is something which is. So, often called as a one time pad, but rather what we do in practice right we try to use the same key for different encryptions.

So, imagine that the same key is used for two encryptions; that means, for encrypting m 0 and as well as for encrypting m 1. So, therefore, right if I take the cipher c 0 and c 1 and if I XOR them; then I know that this k and this k gets cancelled, because k xor k will get cancelled ;and so therefore I will have m 0 XOR with m 1. So, this immediately shows that the notion of difference actually does not depend upon the key and the key gets cancelled.

So, this notion of difference is actually technically called as differential. And therefore, we try to calculate the differential of two states which essentially does not depend upon the secret key.

(Refer Slide Time: 27:28)



An Exercise on Cryptanalysis

x	0 1 2 3 4 5 6 7 8 9 a b c d e f
S[x]	6 4 c 5 0 7 2 e 1 f 3 d 8 a 9 b

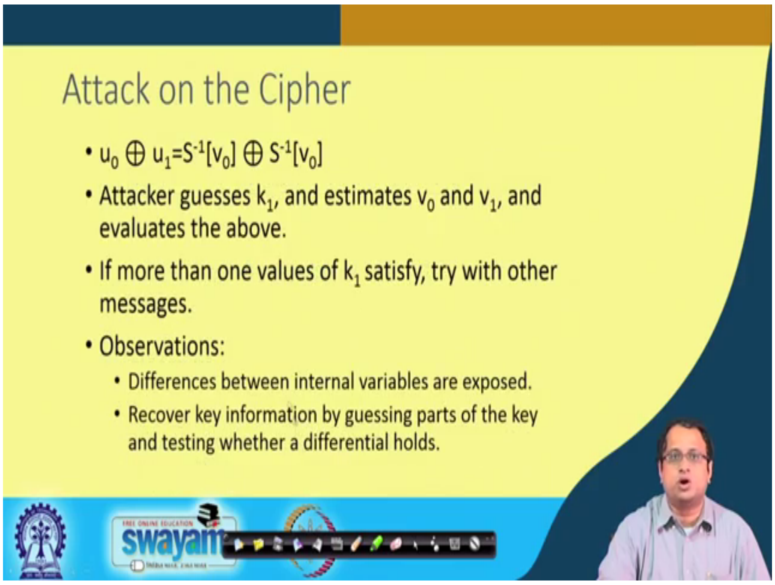
- Study the 'differential' behavior.
- Consider two encryptions with a pair of plaintexts, m_0 and m_1.
- The attacker knows the difference of the internal values, $u_0 \oplus u_1$.

So, let us do a very simple exercise on crypt analysis. So, suppose I have got a message m; I have got this k 0 key and I obtain some intermediate data which is called as u and pass it through an S box. So, the S box is as small as box which I have shown here in the table. So, that just stands for a mapping. That means, if I have got 0 as input then this gets mapped into 6; if 1 is an input it gets mapped into 4 and so on. So, you can imagine that this is just as a lookup table.

So, this S box or S function does that look up and transforms u into v and finally, I apply k 1 to XOR with k 1 and I get c ok. So, you can see that if I study the differential behavior then we can do an interesting attack; rather than you know like guessing all the possible values of k 0 and k 1. So, what essentially you can do here. So, you can observe that k 0 is a 4 bit of information; k 1 is also a 4 bit of information. So, a brute force search would therefore, require 2 to the power of 4 into 2 to the power of 4; that means 2 to the power of 8 possible guesses.

So, we can do better than that and as I will show here in this these two these slides; so suppose you consider encryptions with a pair of plaintext m 0 and m 1; so therefore, as I said in a previous slide, if you know m 0 and m 1; you also know u 0 XOR u 1 because the k will have no effect on that differ on the differential.

(Refer Slide Time: 28:44)



Attack on the Cipher

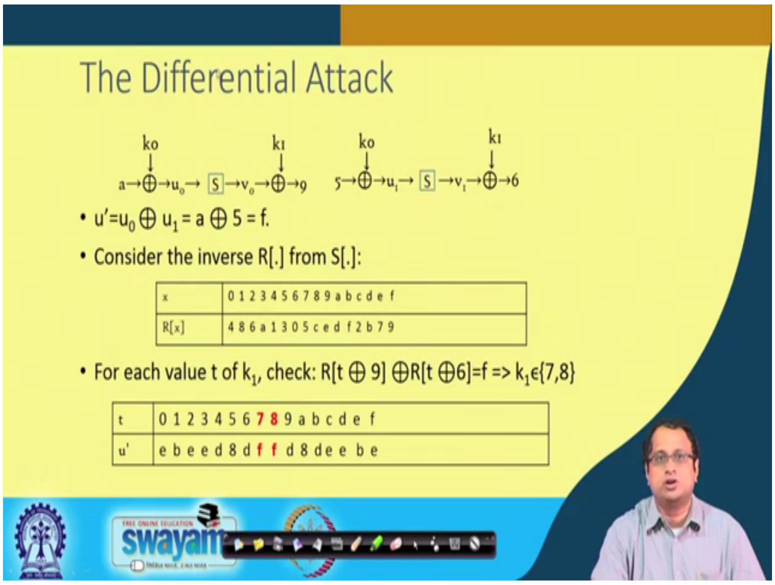
- $u_0 \oplus u_1 = S^{-1}[v_0] \oplus S^{-1}[v_0]$
- Attacker guesses k_1, and estimates v_0 and v_1, and evaluates the above.
- If more than one values of k_1 satisfy, try with other messages.
- Observations:
  - Differences between internal variables are exposed.
  - Recover key information by guessing parts of the key and testing whether a differential holds.

So, once you observe u 0 XOR u 1; you also see that u 0 XOR u 1 is nothing, but S inverse v 0 XOR with S inverse of v 1 ok.

So, you see that if I just go back here then this XOR; the differential here can be obtained by the by XOR in S inverse of v 0 with S inverse of v 1. And so therefore now what the attacker will do is that the attacker will guess k 1 and we estimate v 0 and v 1 and evaluate the above ok; if more than one value of k 1 will satisfy, then it will try with other messages.

So, the observation is that the difference between internal variables will get exposed because of the difference and weaker and this will help us to recover key information by guessing parts of the key and testing whether the differential holds. So, note that when you are guessing here you are just guessing for k 1; it is independent of the guess of k 0. So, you are doing an analysis which does not depend upon k 0 and you can only constant on one part of the key; so you can do a divide and conquer analysis ok.

(Refer Slide Time: 29:45)



The Differential Attack

$$a \rightarrow \overset{k_0}{\oplus} \rightarrow u_0 \rightarrow \boxed{S} \rightarrow v_0 \rightarrow \overset{k_1}{\oplus} \rightarrow 9 \qquad 5 \rightarrow \overset{k_0}{\oplus} \rightarrow u_1 \rightarrow \boxed{S} \rightarrow v_1 \rightarrow \overset{k_1}{\oplus} \rightarrow 6$$

- $u' = u_0 \oplus u_1 = a \oplus 5 = f.$
- Consider the inverse R[.] from S[.]:

x	0 1 2 3 4 5 6 7 8 9 a b c d e f
R[x]	4 8 6 a 1 3 0 5 c e d f 2 b 7 9

- For each value t of k_1, check: $R[t \oplus 9] \oplus R[t \oplus 6] = f \Rightarrow k_1 \in \{7,8\}$

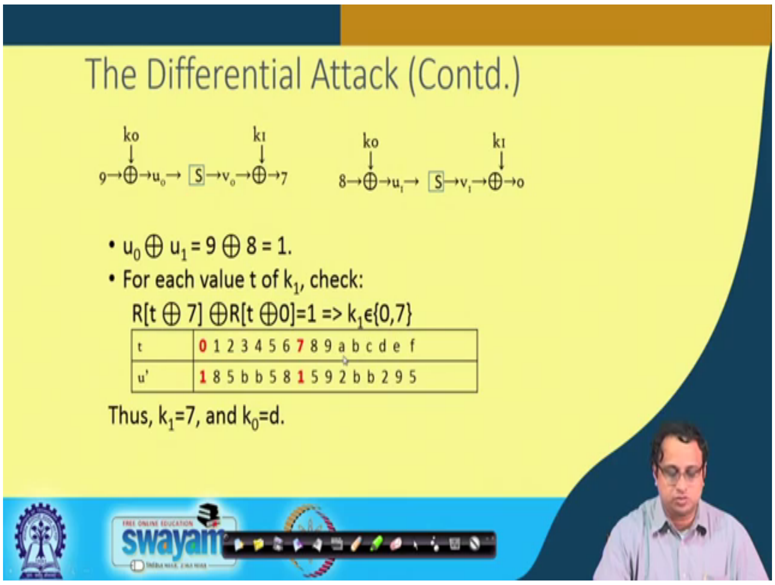
t	0 1 2 3 4 5 6 **7** **8** 9 a b c d e f
u'	e b e e d 8 d **f** **f** d 8 d e e b e

So, here is the; you know like the actual differential attack. So, here you can see what we have done here is that you have got a, which is one of the message, and you have got k 0 you have got u 0 and so what I mean other in the other paired we choose 5 as the message ok.

So, if you exhort between a and 5; then you get f and therefore, u 0 XOR with u 1 is also equal to f ok. So, usually XOR with u 1 is equal to a XOR with 5 and that is equal to f. So, now if you obtain the reverse S box which you can also write because it is a bijective maps; so you can rather than expressing v 0 in terms of u 0; you can also express u 0 in terms of v 0. And therefore, this is the corresponding mapping which you can analyze ok; which you can test for its validity given the previous S box and now you can observe that for each value t of k 1; we want to ensure that r t XOR with 9 ok.

So, is here you can see that 9 is the cipher; so 9 XOR with k 1 and if I apply the r function I come to this point u 0. So, I XOR R t XOR with 9 and here I XOR R t XOR with 6; t is the candidate for k 1. So, I XOR t with 6 and then I apply the inverse function which is R. And then if I XOR then I should get used it or XOR with u 1 which is f ok. So, here you can observe that if I see the inverse function; then there are two t values shown as 7 and 8; for which I get these two fs and therefore, 7 and 8 are my candidate keys for k 1 ok.

(Refer Slide Time: 31:21)



The Differential Attack (Contd.)

$$\begin{tikzcd} & k_0 \arrow[d] & & & k_1 \arrow[d] & \\ 9 \arrow[r] & \oplus \arrow[r] & u_0 \arrow[r] & \boxed{S} \arrow[r] & v_0 \arrow[r] & \oplus \arrow[r] & 7 \end{tikzcd}$$

$$\begin{tikzcd} & k_0 \arrow[d] & & & k_1 \arrow[d] & \\ 8 \arrow[r] & \oplus \arrow[r] & u_1 \arrow[r] & \boxed{S} \arrow[r] & v_1 \arrow[r] & \oplus \arrow[r] & 0 \end{tikzcd}$$

- $u_0 \oplus u_1 = 9 \oplus 8 = 1$.
- For each value t of k_1, check:

$R[t \oplus 7] \oplus R[t \oplus 0]=1 \Rightarrow k_1 \epsilon \{0,7\}$

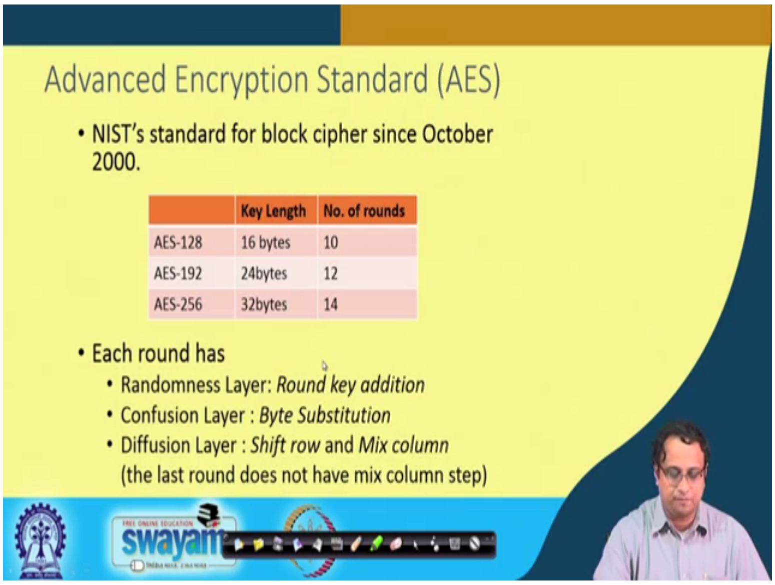
t	0 1 2 3 4 5 6 7 8 9 a b c d e f
u'	1 8 5 b b 5 8 1 5 9 2 b b 2 9 5

Thus, k_1=7, and k_0=d.

So, then I repeat this process for another such pair. So, here for example, I have got it for 9 and 8 and I again repeat this process ok. So now, I have got two key values for k k k 1 which is shown as 0 and 7. So, now, if I take an intersection if we do in previous one like 7 and 8 and between 0 and 7; then I know that 7 is my candidate for k 1 and then I can calculate k 0 is d ok. So, you see that the brute force search here is now was 2 power of 8.

But I am doing this attack with a complexity of 2 power of 4 and that is definitely better than a brute force search ok; so therefore, this is an attack.

(Refer Slide Time: 32:02)



Advanced Encryption Standard (AES)

- NIST's standard for block cipher since October 2000.

	Key Length	No. of rounds
AES-128	16 bytes	10
AES-192	24bytes	12
AES-256	32bytes	14

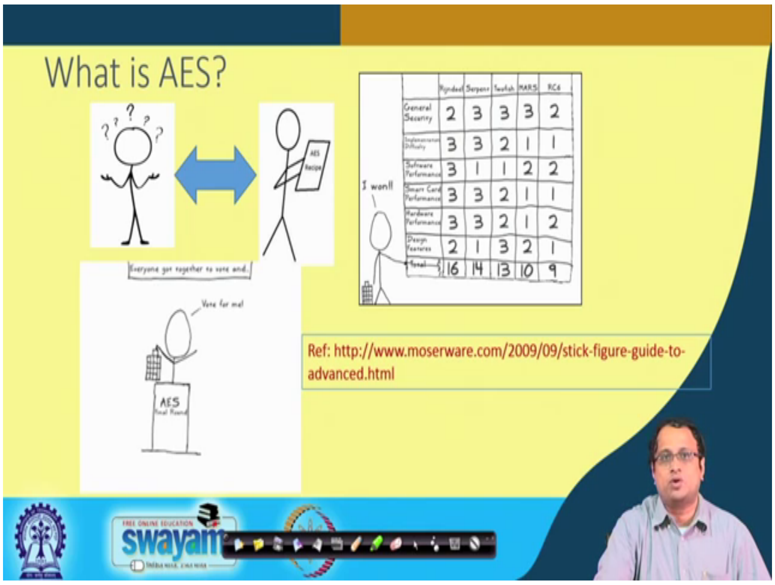
- Each round has
  - Randomness Layer: *Round key addition*
  - Confusion Layer : *Byte Substitution*
  - Diffusion Layer : *Shift row* and *Mix column* (the last round does not have mix column step)

So, talking about you like what we will be talking subsequently is that we will talking about a full-fledged cipher which is called as AES which was that which is also called as the Advanced Encryption Standard. So, this is a de facto worldwide standard from 2000 and there are different versions of this algorithm. So, there are AES 128, AES 192 and AES 256.

So, the idea is that all of them have got 128 bit block will be bit of plaintext, but they have got different sizes of the keys. For example, AES 128 has 16 bytes and it operates on 10 rounds. And likewise AES 192 operates on 24 bytes and so on and it has got a number of rounds is 12. And likewise AES 256 operates on 256 bytes; 256 bits of key and has got fourteen rounds for its encryption.

Each round has got different steps as I said for doing the exhorting with the key and that is typically done by round; round key addition. It also does several you know like functions for confusion which is often done by a process which is called as byte substitution which we will see in the next classes. So, it also does a interesting diffusion it has an interesting diffusion layer which is done by shift row and mix columns and the final round does not have mix columns because it does not add to any conventional security.

(Refer Slide Time: 33:18)



What is AES?

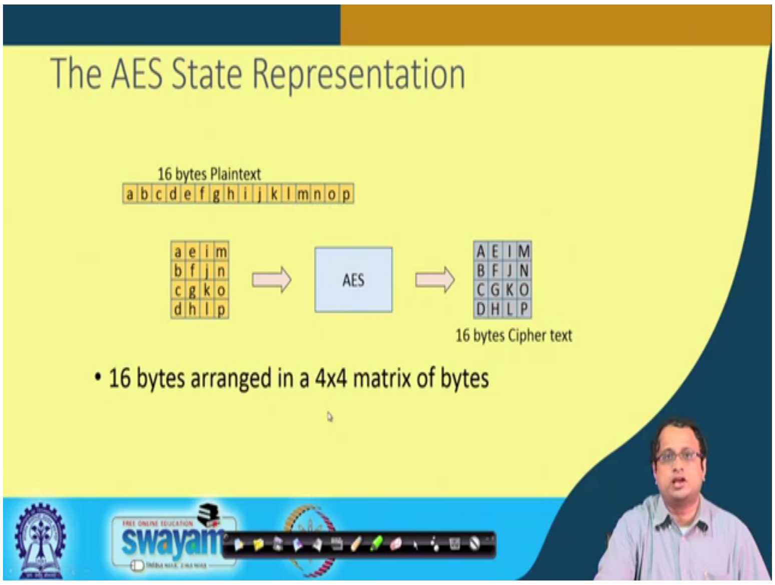
	Rijndael	Serpent	Twofish	MARS	RC6
General Security	2	3	3	3	2
Implementation Difficulty	3	3	2	1	1
Software Performance	3	1	1	2	2
Smart Card Performance	3	3	2	1	1
Hardware Performance	3	3	2	1	2
Design Features	2	1	3	2	1
Total	16	14	13	10	9

Ref: http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html

So therefore, right what is AES? We will be you know looking into this in the next class. And there are like they were I would like to mention here that in the original proposal for AES. There were different algorithms which we were you know like proposed, but (Refer Time: 33:32) or there is not as was the name of the original algorithm that won over several other algorithms because of not only security, but also because of implementational advantages ok.

So, this is this particular cartoon or stick diagram I have taken from this you know like this URL. So, I would like to thank for that.

(Refer Slide Time: 33:55)



The AES State Representation

16 bytes Plaintext

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p

a	e	i	m
b	f	j	n
c	g	k	o
d	h	l	p

AES

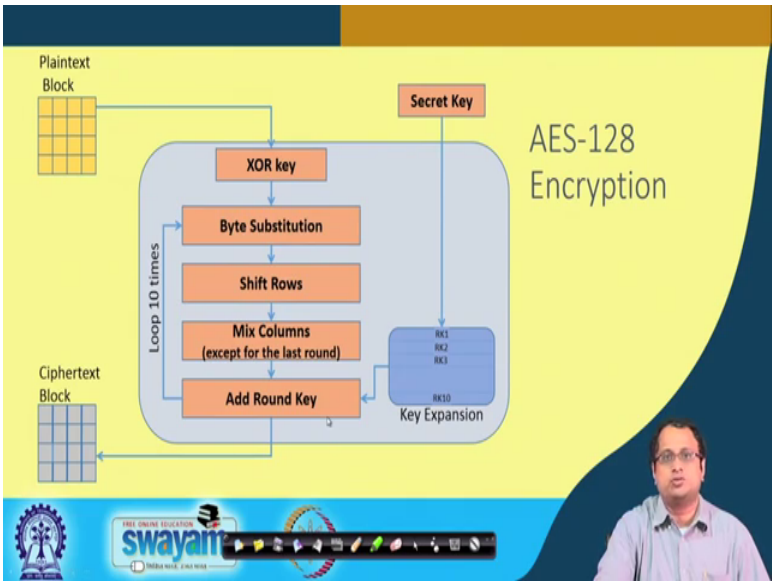
A	E	I	M
B	F	J	N
C	G	K	O
D	H	L	P

16 bytes Cipher text

- 16 bytes arranged in a 4x4 matrix of bytes

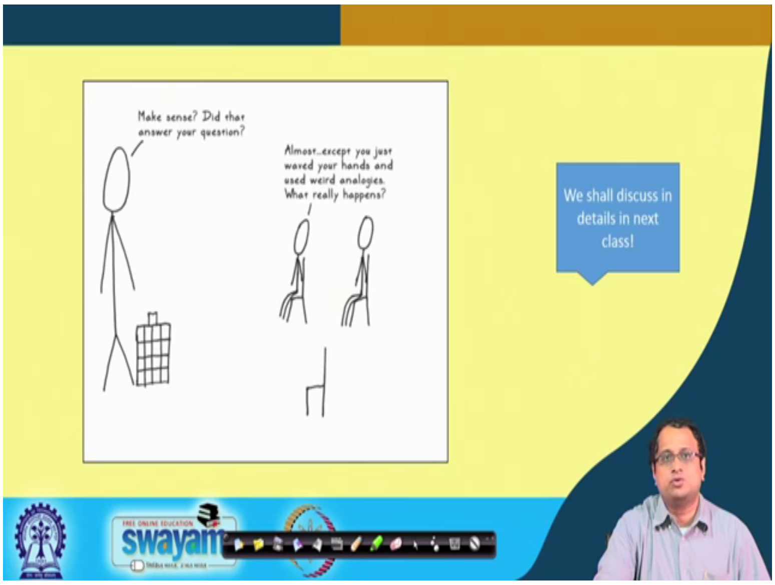
And finally, right we will try to take a look into how the AES looks like and that is something that we will be trying to study more in more details in the next classes ok.

(Refer Slide Time: 34:01)



AES-128 Encryption

(Refer Slide Time: 34:06)


Make sense? Did that answer your question?
Almost...except you just waved your hands and used weird analogies. What really happens?

We shall discuss in details in next class!

So here I would of course, like I have tried to give you an overview behind this cryptographic algorithms, but we will be studying about you know like AES in more details in the next classes ok.

(Refer Slide Time: 34:18)

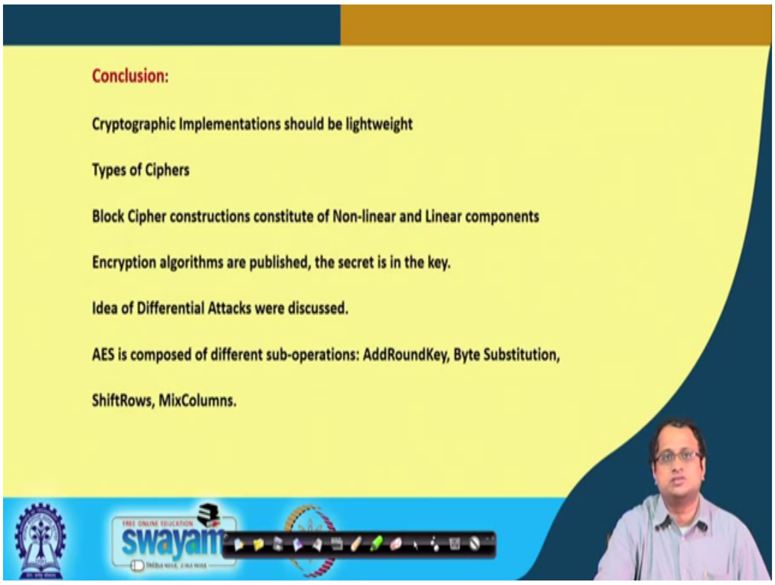
References

References:

❑ Debdeep Mukhopadhyay and Rajat Subhra Chakraborty, Hardware Security: Design, Threats and Safeguards, CRC Press

So, this is my reference for the book.

(Refer Slide Time: 34:22)



Conclusion:

Cryptographic Implementations should be lightweight

Types of Ciphers

Block Cipher constructions constitute of Non-linear and Linear components

Encryption algorithms are published, the secret is in the key.

Idea of Differential Attacks were discussed.

AES is composed of different sub-operations: AddRoundKey, Byte Substitution,

ShiftRows, MixColumns.

And so what we basically studied is we tried to understand that cryptographic implementations should be lightweight, because we need to ensure that our designs have less overhead. We studied about different types of ciphers or broad types of ciphers. We studied about block cipher constructions constitutes of both non-linear and linear components.

That means, which produces the confusion and the diffusion respectively. We studied about the fact that encryption algorithms are published; the secret is often in the key. And we discussed about an idea of differential attacks we also discussed that AES is composed of different sub operations add round key and byte substitution which we will take up in more details in the next class ok; so add round key, byte substitution, shift rows and mix columns.

So, these are different operations which you will try to study in the next class ok. So, with this I would like to say.

Thank you.