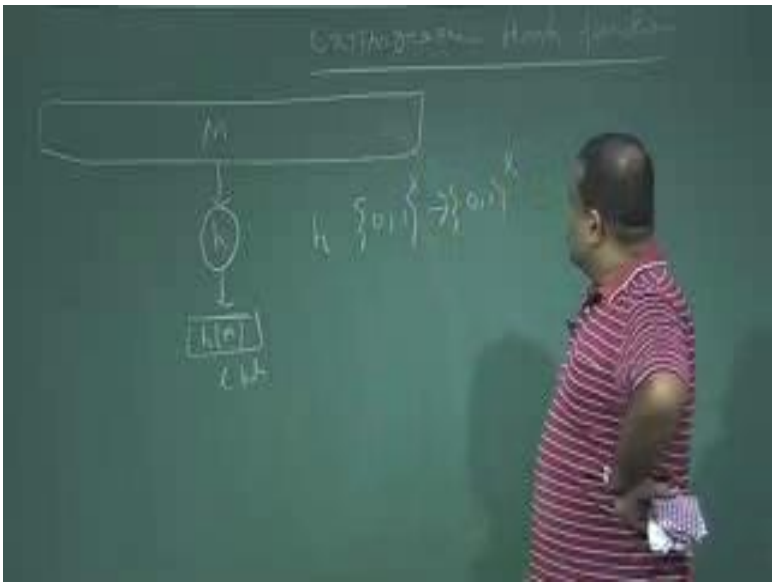
Internetwork Security
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture - 42
Cryptographic Hash Function

So, we will talk about cryptographic hash function, how the hash function is used in cryptography and what should be the desirable property of such function, we should have in order, in the domain of cryptography. So, for that we need to know how we can use this hash function.

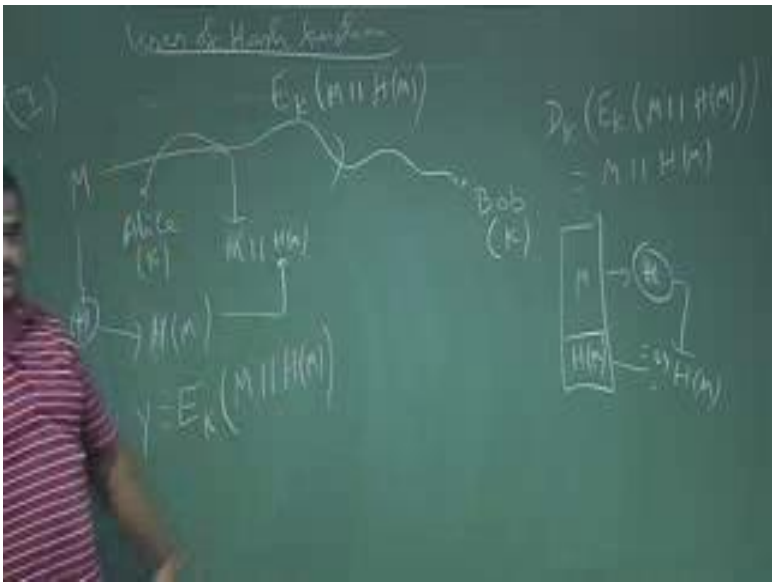
(Refer Slide Time: 00:45)



So, basically in cryptography we use the hash function to reduce the size of the messages, because message size could be big any arbitrary length. So, this is a big message. So, the idea is to reduce to a fixed size say l bits, so that is why we use the hash function.

So, hash function is basically should be able to take any arbitrary length and it should give us the fixed bit plaintext which is called hash digest, so that is the thing. So, before knowing the what should be the property this hash function should have for our cryptographic cells, so before that we need to know what are the uses of this hash function in cryptography, then we can decide what the property should our hash function have. So, basically hash function is used for authentication purpose.

(Refer Slide Time: 01:46)

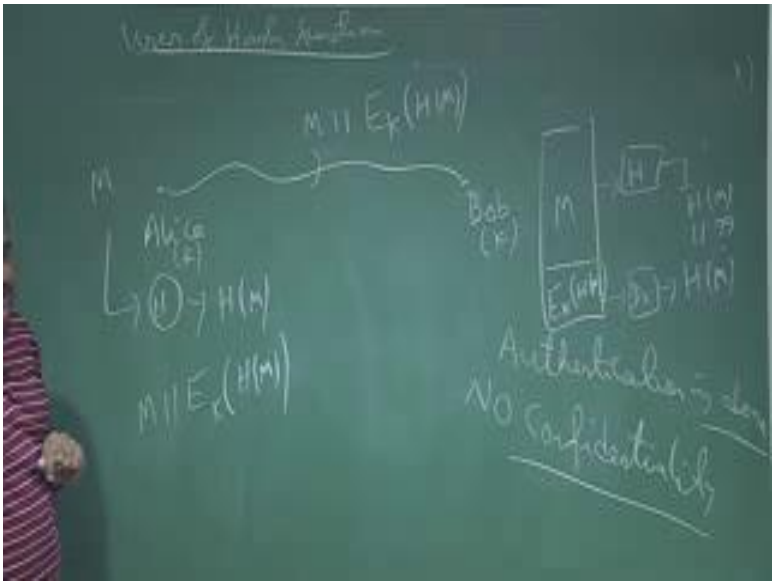


So, suppose we have two parties Alice and Bob. So, this could be say first use. So, uses of hash function in cryptography. So, this is Alice and Bob they are in a symmetric key setup say so they have a common key k, symmetric key and they want to authenticate. So, Alice is having a message - a big message m which Alice wants to send to Bob; and Bob has to ensure that the message is really coming from Alice, so that is what is the authentication. So, what Alice will do? So, Alice will first apply the M on this. So, Alice will first apply the hash function on this message sorry h and get the h of M and then Alice will concatenate this message along with this h of M, and then Alice will encrypt this using the key. So, this is a symmetric key encryption, it could be a block cipher or stream cipher it could be DES, AES or any other stream cipher. So, anyway this is symmetric key encryption and this is the Y. So, this Alice will send it to Bob over the public channel this Y. So, basically Alice is sending E k of N H of M.

Now, how Bob will authenticate this? So, Bob is getting this message this is encrypted things. So, Bob will decrypt it using the key K Bob is having, so H of M. So, this will cancel out if Bob is getting this thing. So, Bob is basically receiving now M and H of M. So, now what Bob will do? So, Bob will apply the hash function, hash function is public function. So, Bob will apply this hash function. So, anyway sometimes we should follow the same convention either capital H or small h. So, Bob will use the same hash function capital H and getting the H of M and Bob will check whether these two value are same or not. If these two value are same then Bob will ensure that the message is coming from

Alice. So, authentication is done, authentication. So, if these two are matching then the message is coming from Alice. So, this is one use.

(Refer Slide Time: 05:36)



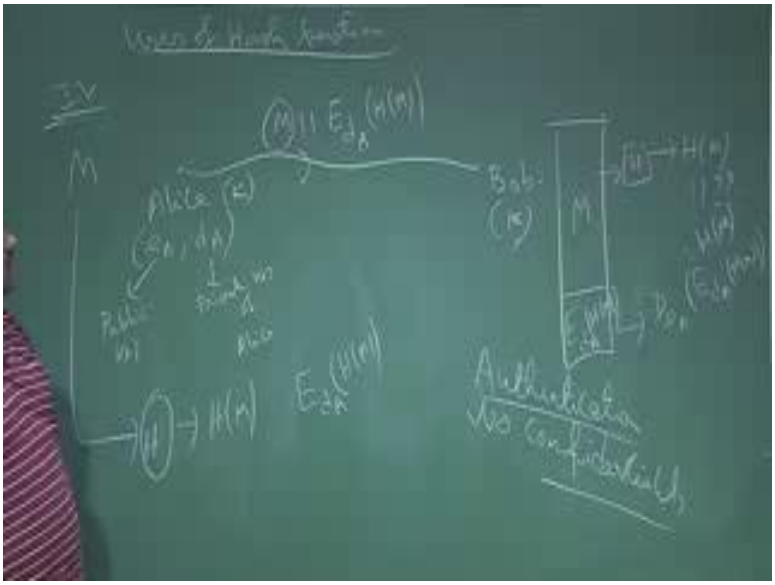
Now, the second one is so next use is say so what we are doing. So, Alice having the message, Alice wants to send it to Bob. So, what Alice is doing? Alice is applying this H hash function and generating the H of M and then H of M is reduced size. So, this is also symmetric key setup. Now, Alice is in the symmetric key they have the common key k. So, now Alice is encrypting this, because this is of lesser size. So, Alice is encrypting this using this k. So, this is symmetric key encryption it could be block cipher or stream cipher then Alice is padding this I mean just appending and send it to Bob.

Now, this is Bob is receiving. So, basically Bob is receiving this M along with this E k of H of M. So, what Bob will do? Bob will first decrypt this, this part decryption and Bob will get back H of M. So, Bob has to know the length of the message because otherwise Bob will be not knowing what is the ciphertext has to decrypt. So, the length of the message is also to be known should be known to the Bob. So, it will just decrypt it and it will get H of M. So, now, Bob will apply H function on this. So, this is also supposed to be H of M, Bob will compare these two value, if these two values are same then Bob will ensure that the message is coming from Alice. So, the authentication is done.

But here we are losing the confidentiality of the message, because M we are sending over the public channel. So, in this scheme no confidentiality of the message, because we

are just sending the message over the public channel, we are just appending the message with this, because Alice has to. But earlier case when we encrypting message and the H of M, there we are achieving both the confidentiality and the authentication. So, this is the symmetric key setup. Now, we can similar thing we can have in public key setup. So, this is use number 2.

(Refer Slide Time: 08:32)



Now, we will talk about use number 3 of hash function. So, this is in public key setup. So, Alice and Bob now, they are in public key setup so that means, they are having public key; they are having two pair up keys. So, Alice is having e of A, d of A, this is basically public key of Alice and this is basically secret key or the private key of Alice. Now, Alice has a message M which is really big message and Alice wants to send this message to Bob in a way that Alice wants to authenticate that I am sending. So, Bob has to ensure that the message is coming from Alice that is the authentication. So, what Alice will do Alice will apply the hash function on it, and get the message H of M.

Then Alice will apply the signature on this message. So, Alice will do the sort of encryption, but using Alice secret key, because it is basically signature signing Alice is signing on the message. Now, because this is reduced size then it would not be so much expensive because m is very big. So, if we have to do the public key encryption on M on a bigger space, it is more expensive because public key is more expensive than the symmetric key. So, now it is ok if we can so Alice will sign on this. So, Alice will

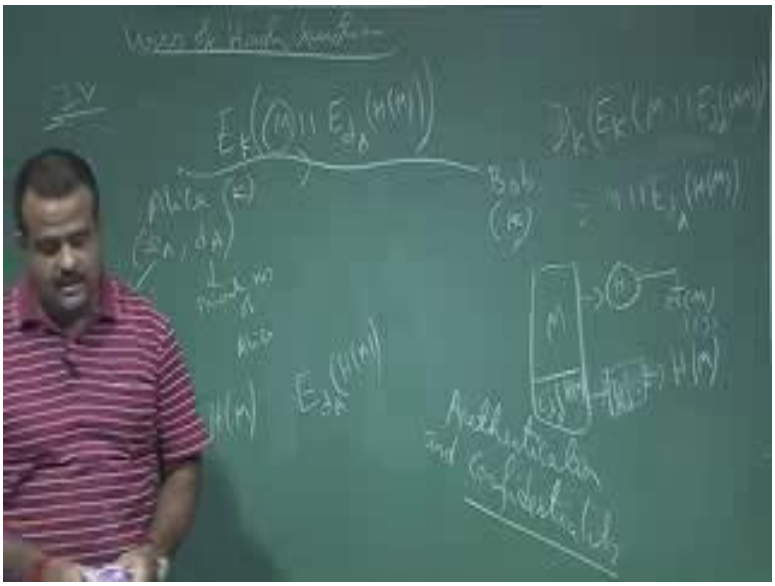
encrypt this using Alice secret key. This is one reduce size and then Alice will send the message along with this signature, and this Alice will sent to a Bob.

So, how Bob will verify? If the message is coming from Alice actually Bob is getting this message along with this Alice signature on this hash value of this message, because it is reduce size. So, what Bob will do? Bob will first get it back that means, Bob has to apply the inverse of this using the Alice public key which is known to everybody though these thing can be done by other party also Oscar also. So, Bob will just decrypt using the Alice public key on this H of M, and this will give us H of M because encryption is the reverse of the decryption. So, this quantity will give us basically H of M and now Bob will compute h on this and Bob is supposed to get H of M. So, Bob will compare these two whether these two is matching or not if these two is matching then the authentication is done then Bob will ensure that the message is coming from Alice. So, here authentication is done.

But here all similar problem no confidentiality, because message is becoming public, sending the message over the public channel, so now, if we want confidentiality here also, so what we do? Suppose on top of this so this is another use, this is number four say. So, they have a common key k, which they have this is symmetric key; they have a key which is shared between them, they agreed with that maybe we can use the Diffie-Hellman exchange key for that. So, they have a secret key k, which is known to Alice and Bob, and that is the secret.

Now what they can do? Now they can encrypt this thing using that secret key, and this they can send to Alice can send to Bob. So, what Alice is doing? Alice is first applying the hash function on it then Alice is signing on the message on the hash value of the message, because this is a fixed length. And then after that Alice is appending this message with this signature, and then Alice is applying this symmetric key encryption this could be any block cipher or stream cipher AES, DES. So, just Alice is applying the symmetric key encryption on the message and send it to Bob.

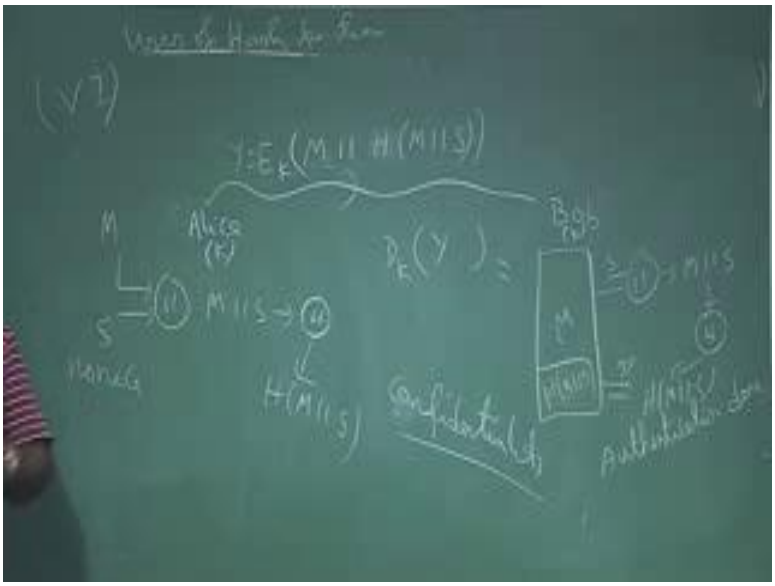
(Refer Slide Time: 14:13)



So, Bob is receiving this now upon receiving this, what Bob will do? Bob is also having a secret key k. So, Bob will decrypt it first D of k - the symmetric key decryption if it is use a AES then AES decryption. So, Bob is getting M along with this E d of A H of M. So, this will give us this will give Bob m along with E D of H of M, so this thing. So, M this is E d of this is the Alice signature. So, again Bob has to do the same thing Bob has to decrypt this. So, this is d of a using the Alice public key E of d of A I mean using the Alice public key. So, this will give us H of M. And Bob is having now M, so what Bob will do Bob will apply this H, and it is also supposed to be H of M, and Bob will try to match this whether they are equal or not. So, this is these way we are if it is matching then Bob will ensure that the message is coming from Alice. So, this is the authentication.

So, here we are having both the authentication and the confidentiality of the message because message is not becoming public we are using another level of encryption symmetric key encryption, so authentication and confidentiality both achieving here confidentiality both are achieving here. Now, in a next use may be like this. So, this is the fifth one. So, here what we do? We will introduce extra security, which is called nonce or some sort of time stamp you can say on the messages.

(Refer Slide Time: 16:29)



So, what we are doing here? We are just we are having, so Alice and Bob. So, Alice is having the message M. So, what we are doing? We are just this is the S concatenation operation we are having a nonce this is a nonce or can be time stamp. So, we got S along with this, and on this will apply the H function. So, this will give us H of this and then this along with the message we are sending to Bob, M along with concatenate with H of M S we are sending to Bob.

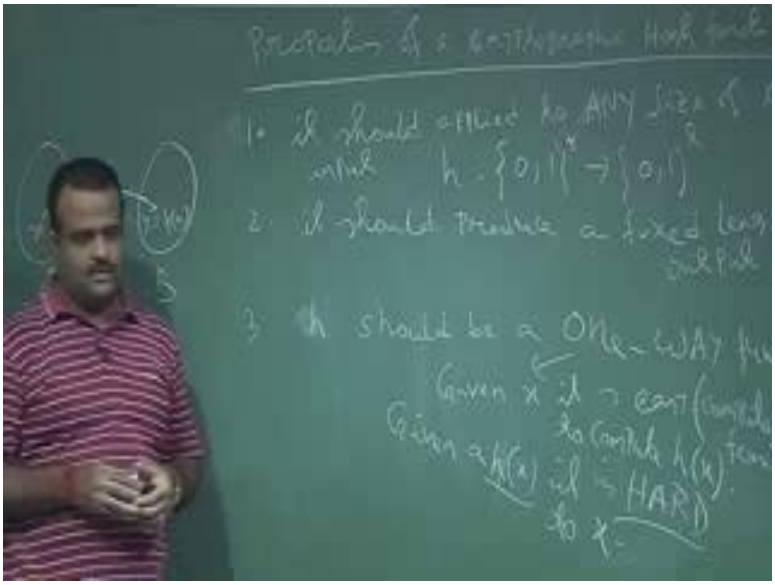
So, now how Bob will get back the message, how Bob will authenticate this messages coming from Alice, so Bob is receiving this thing m along with H of M S. So, what Bob will do? So, this is the concatenation operation. So, Bob will, also should have the same time stamp or same nonce with Bob. So, Bob will do this thing append this S with the message and then Bob will apply the H function on it. So, this should be H of M comma S and this should be equal to this. And if this is equal then the authentication is done, but here also we are losing the confidentiality because the message is becoming public over this public channel.

So, if we in this scheme, if we want to have the confidentiality also that is the sixth one. So, we can similarly we can have a k which is shared between the Alice and Bob, and then what we do? We can encrypt this using this k, and then these we can send it to Bob. So, what Bob will do? First, Bob will decrypt this decrypt this Y and Bob will be getting this and then Bob will do the remaining thing as earlier. So, this way this is the

symmetric key encryption this way we can achieve the both the confidentiality and the authentication.

So, even we can extend this for the public key setup. So, this here we can have the confidentiality also. So, the message is not becoming public, confidentiality. So, the more or less we can think few more application of hash function based on this. So, more or less this is the hash function is used mainly for the authentication purpose, and mainly to reduce the size of the plaintext, so that we can encrypt it because public key encryption is very expensive. So, if we have to encrypt 128 bit long message it is very difficult with RSA or any elliptic curve cryptosystem. So, it is better if you can reduce that so that h of s is helping us to reduce the message.

(Refer Slide Time: 20:20)

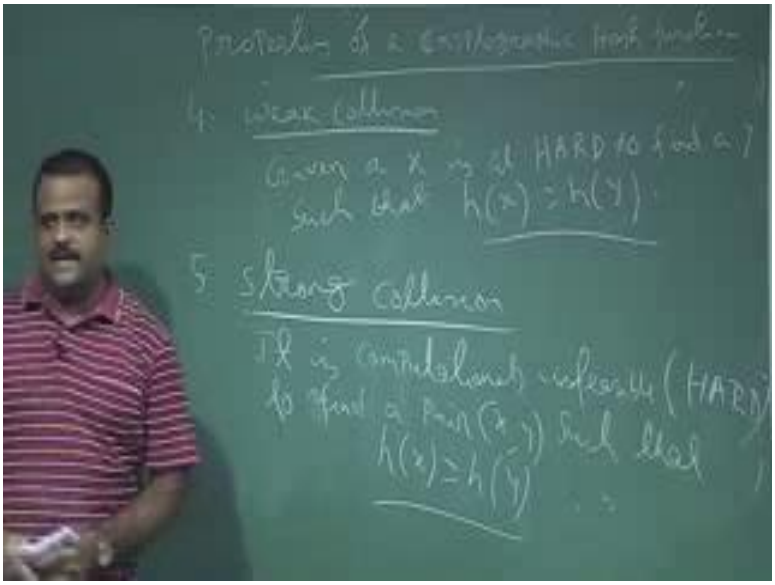


Now, based on the uses now we want to know what are the properties our hash function should have; otherwise we cannot use it in this cryptography. So, what are the cryptographic hash functions should have, the properties? So, properties of a cryptographic hash function should have, first one is it should apply to any size of the input, so our plaintext should be any size. So, it should applied to any size of the input as big as it is. So, basically it is a function from arbitrary length input to a fixed length output that is the first condition. So, it should be applied to any size input, and it should give a fixed size output. So, any size input and it should produce a fixed size fixed length that is l say fixed length output that is l.

And the third condition is it should be a one way function this h should be a one way function. What do you mean by one way function? One way means suppose we have a function f from A to B, this is a mapping f mapping. So, this is x this is y is equal to f of x this is f mapping. So, when we say f is a one way function, if it is easy to compute in the forward direction and it is hard to get back, it is hard to compute in the backward direction.

So, if this part is easy, so given x, so one way means given x, it is easy. Easy means easy is a vague word, easy means computationally feasible so that means, there is a polynomial time algorithm to compute this, it is feasible computationally feasible; that means, a polynomial time algorithm is there to compute this. A given a x it is easy to compute a h of x. So, given a x this part is easy, but this part is hard. Given a f x given a f x it is hard to get hard means again computationally infeasible computationally infeasible hard to get x so that means, it is hard to invert. So, given a f x given a h of x here f is h, basically given a h of x is a hard to get x. So, it is very difficult to invert so that is the hardness, so that is the one way function.

(Refer Slide Time: 24:39)



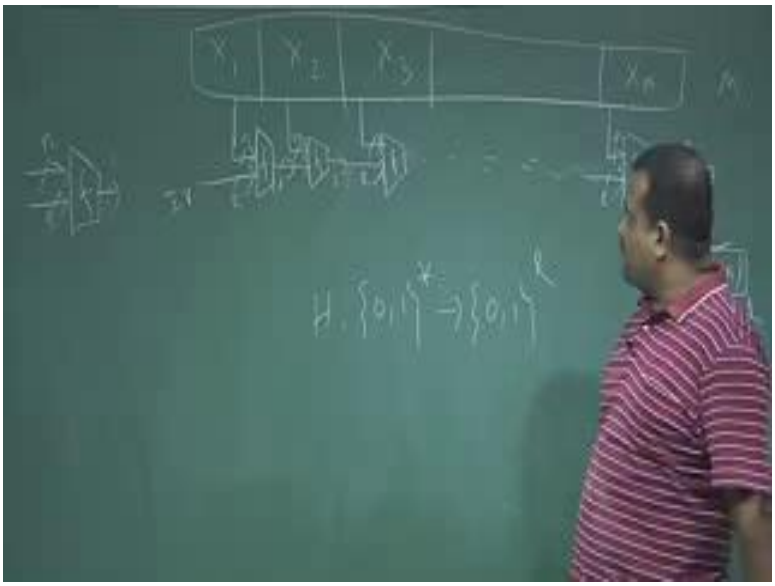
Next property is that is called weak collision property, so that is the number 4, weak collision. It is telling we have given a x, now we need to find a y such that they are colliding. So, given a x, it is hard, hard means again computationally infeasible hard to find a y such that h of x is equal to h of y. So, this is called weak collision property, and

if a hash function satisfying up to this then it is called a weak hash function. So, this property is telling given a x in the domain of key space given a x it is hard to find a y such that they are colliding, this is the slot such that they are colliding. So, h of x is equal to h of y. So, this is the weak collision.

And the strong collision is, strong collision is basically, so it is hard to find out the pair x y which are colliding. It is computationally hard, hard means computationally infeasible that is the hardness so; that means, there is no polynomial time algorithm to handle this. It is computationally hard to get hard to find a pair of keys x, y such that h of x equal to h of y. So, this is the strong collision property. So, if a hash function is satisfying this property, this earlier three property and this property, then it is called a strong hash function. This is strong collision property. Strong collision property means it is very difficult it is hard to get pair x, y such that they are colliding; and weak collision is we have given a x we have to get a y such that they are colliding. So, these are the properties our hash function should have in order to we have to use the hash function. We have seen the uses; for those uses, we need to have these properties. So, any cryptographic hash function should have these properties.

Now, we have to construct a cryptographic hash function. So, general construction of a cryptographic hash function how a hash function can be constructed. Hash function is basically it is reducing it is taking any bit input message, and it is reducing to a fixed bit output. So, this is a general structure of a hash function.

(Refer Slide Time: 28:16)



So, it is taking any bit input. So, this is the big message M, this is the message M. So, what we are doing, suppose we have a function f this is the hard of the hash function f which is taking two input; one is say n bit another is say l bit, and which is giving us l bit output. So, f is basically taking two input n bit cross l bit to l bit. This is the compression function this is the heart of the hash function, every hash function has such function every hash function is having such f. Suppose, we have such a function which is taking a two input, one is n bit another one is l bit, and it is giving us the output l bit. Now how we can use this function to have a hash digest.

So, what we do? This a long message we divide into blocks of size n bit. So, each of this is n bit. So, this is a last block x k. So, we may need to append something to make it n bit. So, what we do? We first apply this f function on this with this n bit, and we have to start with some starting vector that is the IV, and it is producing this is the n bit this is l bit it is producing l bit.

So, again we have to apply this f function which is taking this input, this is also n bit. So, it is l bit again, this is l bit again, so this is f function. So, repeated application of the f function. So, again we do like this. So, again this is f, this is one input, and this is another l bit like this we continue. So, this is one input, this is last block, this is dot, dot, dot, this is our l bit and this is our H of M. This is basically l bit, this is m bit, this is l bit. So, this is the hash digest. So, this is the hash function this is the digest. So, this is H of M. So,

every hash function has this compression function, which is the heart of the hash function, so which is taking two input n bit and l bit, which is giving us output. So, this hash function is a function from 0, 1 any arbitrary bit to 0, 1 to l bit.

So, next class, we will discuss some particular hash function like SHA 1 - secured hash code. So, there we have a particular type of f function that is called m d 4 or m d 5. So, but this is the general structure of any hash function, it has a compression function and we repeatedly applying the compression function and finally, we are getting the hash code of this message M.

Thank you.