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

Lecture - 52
Modern Stream Cipher

(Refer Slide Time: 00:29)

So, we will talk about some different stream cipher some modern stream cipher. So, we have seen in a stream cipher general structure of a stream cipher this is basically there are two types of stream cipher one is synchronized stream cipher. In synchronized stream cipher, what we have? We have a finite state; it could be some state s t, which is initialized by the secret key shared between Alice and Bob, I mean shared between receiver and sender. And then we have the state update function which is taking the, and it is there is a function f which is taking that content of the state and it is giving us the key stream. So, this is the key stream, t-th key stream, and then it is XOR with the plaintext bit p t, and we get the c t ciphertext bit, t-th ciphertext bit within the t-th block.

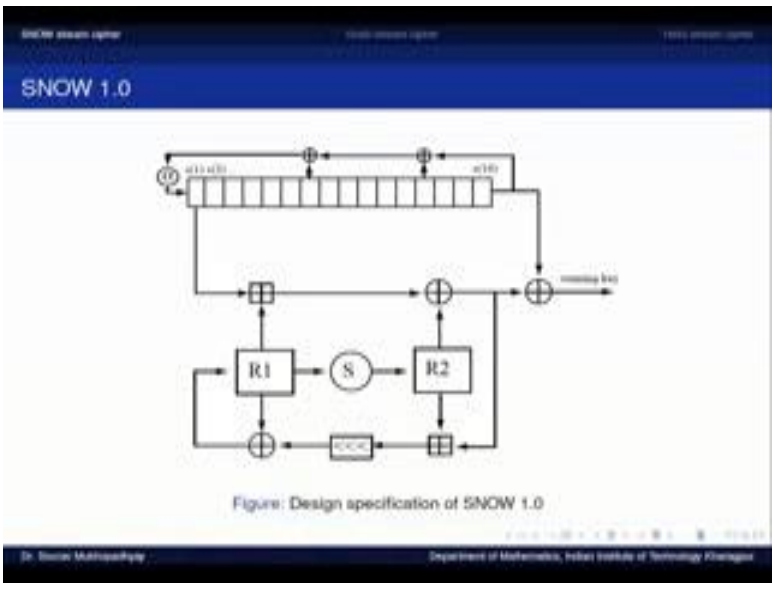
So, in the modern stream cipher, apart from this secret key they are between Alice and Bob, what we have we have the IV - initialization vector, which is also a another secret which is shared between Alice and Bob. So, this IV is used as a, this is called initialization vector this is used in this initialized method of the stream cipher. So, what we do? We take some initialization vector, and we will put it as a part of the key with the

key secret key, we will use this to initialize the state and then we run the key stream for some clocking up to some clocking. And then finally, we will get the initial state of the LFSR, is this clear?

Earlier, the initial state of the LFSR is direct the secret key shared between Alice and Bob. So, now this if we do that then it is a candidate of what is called time memory state of attack. So, to prevent that we want to increase the size of the secret to increase the size of the secret, the IV is introduced. So, we use this IV along with the secret key, and we put it into the state, and we run this state, we clock this state for some number of times, and then we take that after running this state in some number of times then we take that value of that state as initial state. So, that is the every modern stream cipher has this IV, and we put this IV along with the key into the state, and then we run this g the state updation function some number of times and then we reach to the initial state. This is called initial state.

Then after reaching the initial state then we start generating the key stream. This is called initial state; initial state is getting after running this few number of times, after putting this IV and the key into the final state machine, then we clock it many times, and then we stop now this is our initial state. Now, after this we keep on running to get the key stream, so that is the idea.

(Refer Slide Time: 04:19)



Let us talk about some stream cipher, some modern stream cipher which is using the IV and this key stream. So, this is the stream cipher SNOW. So, this is a LFSR, this is basically a 16-bit LFSR, which is connected by this, this bit. And this is s 1 to s 16, and we have a alpha. We will discuss, what is alpha. And then as a part of it, we have a register 32 bit - two register R 1 and R 2 and this is a S-box. And this is a (Refer Time: 04:53) beta XOR operation and this is the addition operation mod 2 to the power 32, and this is the circular shape. So, these are operations in the SNOW 1. And then finally, we will take this bit along with this, and we extort these two and get the key stream bit.

(Refer Slide Time: 05:18)

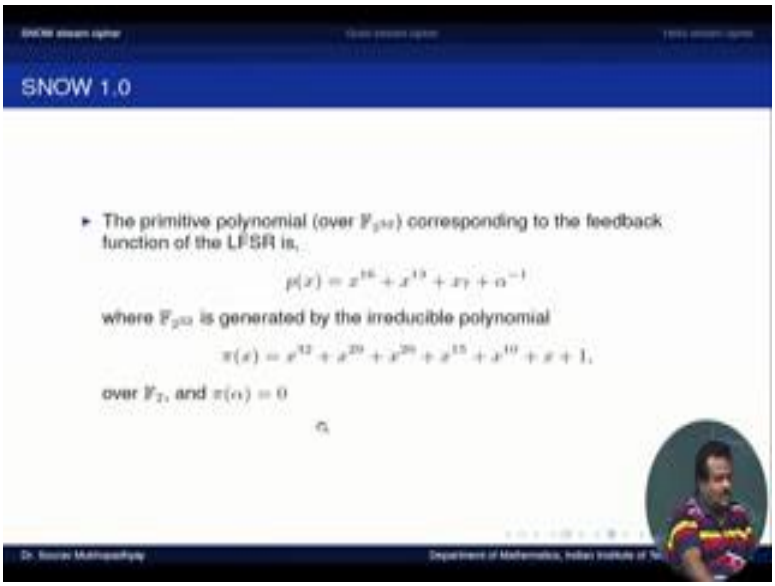
So, this LFSR is 16 bit LFSR and we have a two, this is a, this is also finite state machine. This has two register R 1 and R 2, and operation will be calculated output to the next state the value of by taking the value of R 1 and R 2.

(Refer Slide Time: 05:41)

So, first key initialization is done this procedure provide the initial value of the LFSR as well as for R 1 and R 2. So, we are initializing everything LFSR and the register R 1 and R 2 which was 32 bits.

Next part 32 bit of the key stream is calculated by bitwise adding the output of this finite state machine and the last entry of the LFSR that means this one. So, this is last entry and this is the output of the, what is finally we are telling this as a machine finite state machine this output will be XOR and we are outputting this 32 bits. After that whole cipher is clocked once, and next 32 bits of the keystream is calculated by again bitwise adding the output of the finite state machine and the last entry of the LFSR. We clock again to continue this fashion. So, each time, we are getting 32 bits, 32 bits like this.

(Refer Slide Time: 06:45)

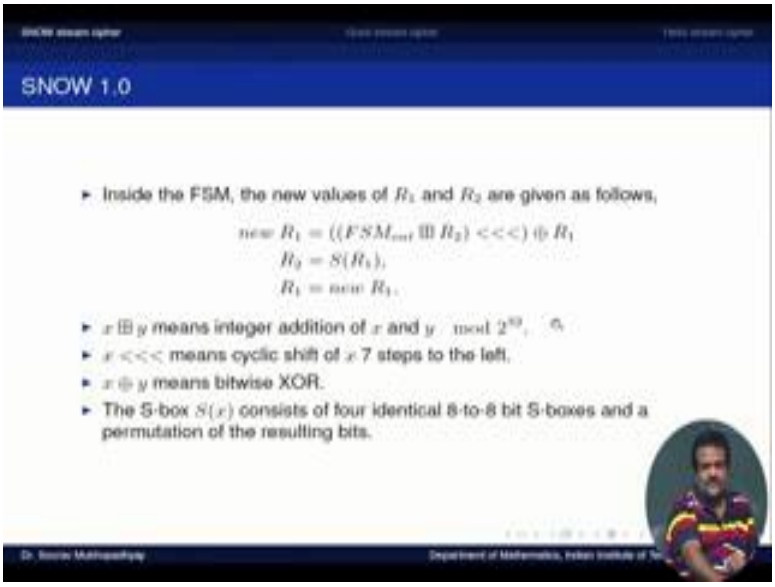


And the LFSR here we are taking here is a primitive based on a primitive polynomial over F 2 to the power 32 and this polynomial is of this form. So, this is 16 bits LFSR, so x 16 plus x 13 plus x to the power 7. So, this x to the power 7, there is a mistake, plus alpha to the power minus 1. This alpha is basically root of this irreducible polynomial. So, this is a irreducible polynomial where F 2 to the power 32, and alpha is the root of this polynomial. So, this alpha you are taking alpha inverse of this. So, this is the LFSR we are choosing so on.

(Refer Slide Time: 07:29)

And these are the state of the LFSR. So, this LFSR is basically flip flop. So, these are the flip flop. And we have output of the finite state machine called FSM out is calculated by this, because it is the s 1 is the S-box which is operating with R 1 this is addition operation and then it is (Refer Time: 08:00) XORing with the R 2, and we are getting the output of this. And this output we have basically XORing with this is again 32 bits, this output here basically XORing with this with the s 16. And the key stream is finally, XORing with the plaintext to produce the ciphertext that we know.

(Refer Slide Time: 08:24)

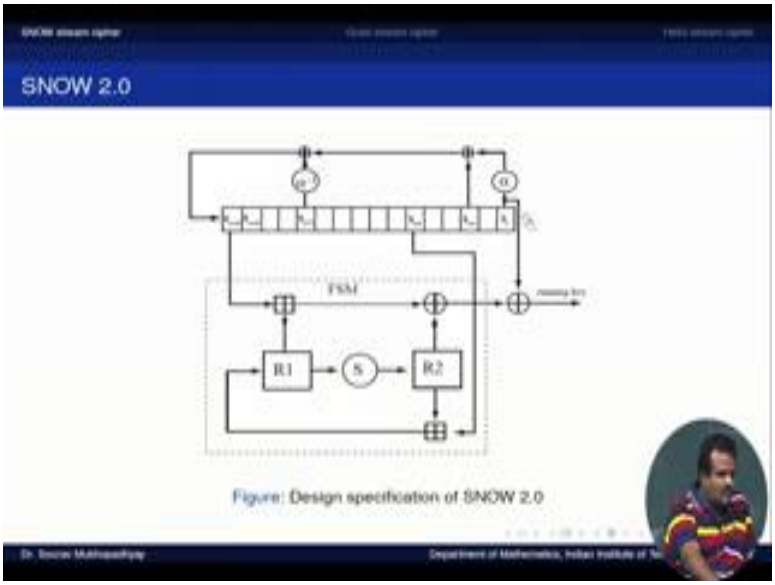


So, inside this finite state machine, we take a value R 1 and R 2, but how this state is changing in the finite state machine. So, let us come back to this figure. So, how this states are changing here. So, this is the state updation function in the finite state machine. So, it takes this R 2 and then this is the shifting circular shifting, we will come to that then it is taking R 1 then XORing with that then it is changing the this R 1. So, then again it is changing the R 2. So, this is what we are doing here.

So, this is the state update function for this state R 1 and R 2. So, this R 1 is basically f of this output we are adding with this is the addition modulo 2 to the power 32. This is we are adding with R 2, then this is the cyclic shift x 7 times. And then final result we are XORing with the R 1, so this is basically updation of R 1. And the R 2 is basically simply we have a S-box, if we come to the figure again, this is basically S-box which is taking the R 1 as input and it is R 2 it is updated by this. So, this is the state updation

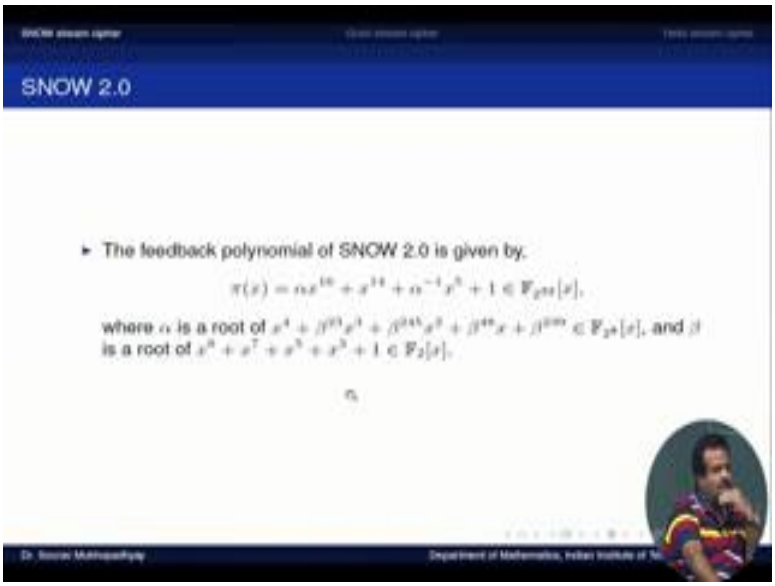
function for this finite state machine. So, this was the attacked by some algebraic attack on many cryptanalysis on the SNOW 1 is there in the literature.

(Refer Slide Time: 10:24)



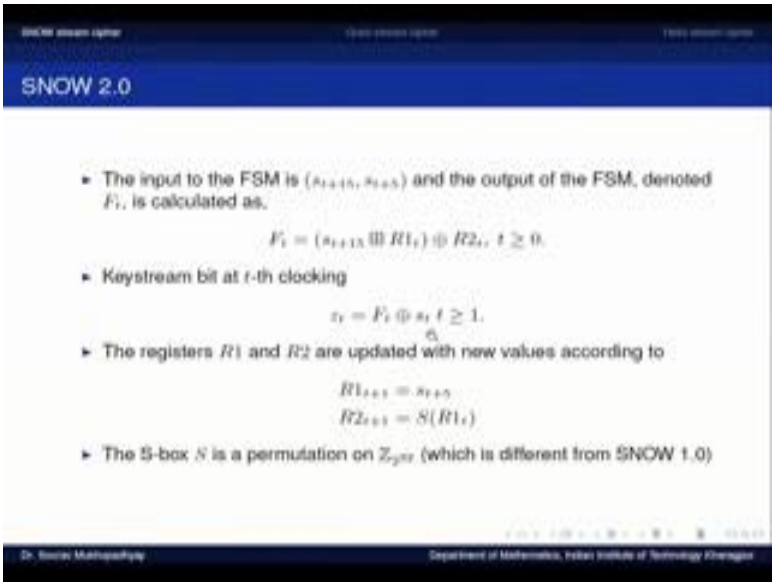
So, they propose a new version of the SNOW that is called SNOW 2. So, in there, they have changed some the polynomial and their connections with instead of taking alpha in box, they have introduced alpha multiplied with alpha inbox. So, this is the SNOW 2. So, remaining are almost same.

(Refer Slide Time: 10:25)



So, except this is coming from this polynomial they have changed the feedback polynomial of SNOW 2 is given by this. So, they are multiplying alpha with this then this positions also they are changing, and then alpha inverse x to the power 5 plus 1. So, this is the polynomial they have used for this SNOW 2, where alpha is the root of they are taking a new polynomial root of this polynomial. So, if I take this polynomial this is a polynomial over F 2 to the power 8, and they have also beta, beta is the root of this x I think this is the polynomial they have used in here.

(Refer Slide Time: 11:05)

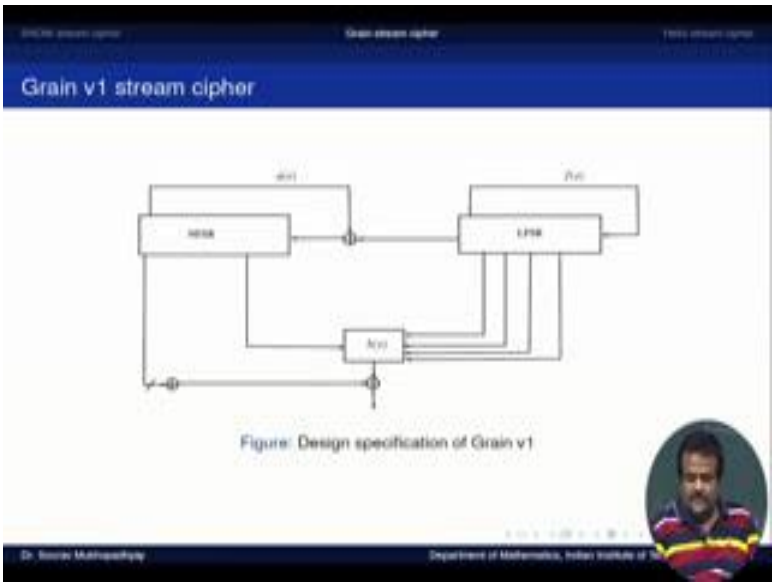


So, they have changed this remaining key streams are almost same; only thing is that position we are changing. So, this is the input, if the finite state machine is this so, but this is how we update the state finite state machine that R 1 and R 2. So, this s t plus 15 and s t plus 5, then it is basically here with s t plus 15, we are adding with R 1 t that is for t-th clocking we are doing this. This is the addition modulo 2 to the power 32 and then we are XORing with R 2 t, at the t-th time, this is the propounding position of this is the state this is the value in the R 2 t register. And then the keystream bit is basically F t XOR with the s t.

So, this is if we come to this figure, so this is the output, this is basically coming from the LFSR bit s t this is XOR with the output. So, this is our keystream bit. So, this is our F t, F t is the output of the finite state machine and then s t which is coming from the LFSR and these two are XOR to get the z t. The register R 1 and R 2 are updated like

this by the similar way earlier. So, this is the S-box we are applying. So, if we come back here, so we are applying the S-box to update this. So, R 1 will be same as this s t, this is F t is basically added with the R 2 and that will give us the new R 1, so that is the state update. And this R 2 basically you take the R 1 and you apply this S-box and that will be the new R 2. So, this is the way how we update this. So, this is the new version of the SNOW, this is called SNOW 2.

(Refer Slide Time: 13:19)



Now, we talk about another stream cipher, which is called grain stream cipher. So, the grain has two versions. So, this grain is having two parts; one is LFSR, another one is NLFSR. So, this is each of NLFSR means non-linear feedback shift register and this non-linear this is LFSR you know feedback shift register, this is non-linear feedback shift register. The connection is I mean this output is non-linear output.

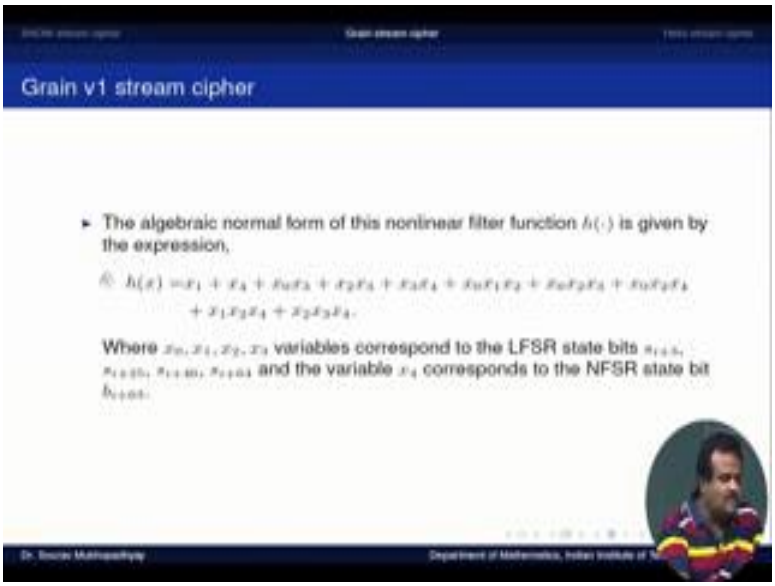
So, this each of this is 80 bit. So, this is a 80 bit LFSR, and this is also a 80 bit LFSR. And this non-linear - this NLFSR is update by the function g and the LFSR is updating by the function F. And to get the key stream you have h function which is taking four bit from this LFSR which four we will come to know 4 bit. So, each have 80 bits. So, among these 80 bits, we are only taking 4 bits from the LFSR and we are taking 1 bit from this NLFSR and we are applying this function h and then the output we are XORing with the NLFSR and that is the keystream. So, this is the general structure of the of the grain stream cipher, and this is basically what we are doing here.

(Refer Slide Time: 14:56)

So, let us talk about this. This is how the feedback function of LFSR is happening. So, it is basically taking this many bits. So, this is a 80 bits. So, among this 80 bits, we are taking the 64 bits, 51 bits, 38 bits, so these are the bits contributing with the this F function. So, this is the function how we making the feedback. So, this is a state updation function of the LFSR. And this is corresponding state of this is the state update function for the NLFSR this is this g function. So, NLFSR is what we are doing we are getting the state form LFSR and this is also XORing with this g of x. So, g of x is taking this input of the LFSR state and apply g function and then it is XORing with the this x t, which is the state coming from the LFSR and then we are XORing from that that is the new bit for the NLFSR. So, this is what we are doing here.

So, this is coming from the LFSR, and this is the way this is our g function. So, g is involved many bit from the NLFSR. So, that is 62, 60 f bit, 52 bit, 45 bit like this. So, this is just all the bits are involved, and this is not a linear function. If you observe here there are some non-linear components over here. So, this is the multiplication I mean this is also four five four variables are involved. So, that is why this is called non-linear feedback function. So, that is our g function, this g function. So, g is taking few state of the NLFSR, and it is a non-linear function and then it is XORing with this LFSR function and it is going to the NLFSR bit. So, this is the update function of this is the bit which is updating the NLFSR.

(Refer Slide Time: 17:22)



And then we have the h function. So, h function is taking, so if you this is the function which is giving us the key stream h of x. So, what is h, h is basically let us go back to the picture. So, h is taking how many bits, h is taking four bits from the LFSR and one bit from the NLFSR and that is giving a that is the input of the h and h of x is the output here, so 4 bits from the LFSR and 1 bit from the NLFSR. So, which four, so this is our h function, so x 0, x 1, x 2, x 3 those are coming from NLFSR. So, these are basically combination of these and the remaining bits are basically, and x 4 is coming from NLFSR and this is the h x function. So, this keeps on updating this.

So, basically if the LFSR are which bit for 3rd bit, 3rd position, 25th position and 46th position and 64th position. So, these are the positions coming from the LFSR bit contributing in the h function. And for NLFSR, it is basically 63th position, so 63. So, this is 8 bit among this 8 bit, so 63th bit is basically our one of the input in h, so that is basically our x 4. And this is the function of this, this is also non-linear function here also non-linear function over here.

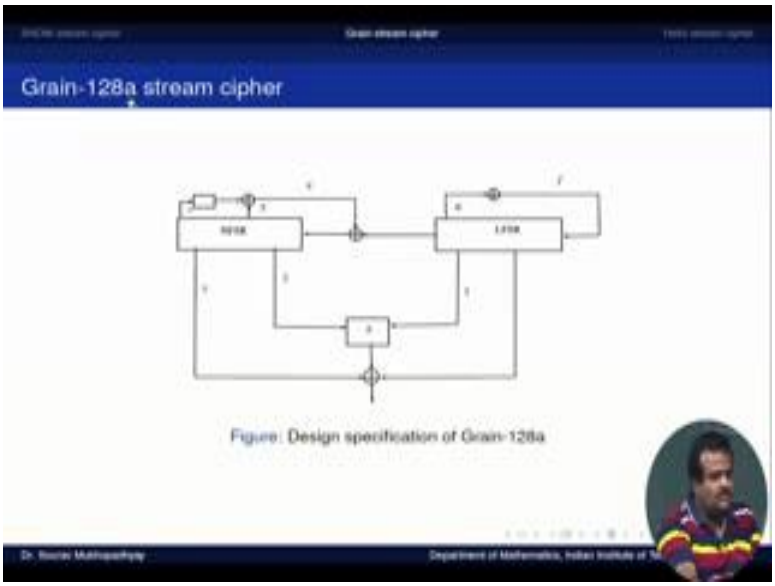
(Refer Slide Time: 19:08)

So, this is making as z t. So, z t is basically algebraic form, so algebraic normal form of the keystream bits that is ciphertext after clocking this. So, now we have to talk about the initialization state of the LFSR I mean this grain. So, how we because we know this stream cipher is of we have this finite state machine and then we have this Alice and Bob sharing the secret key along with the IV. Now, we put this IV in the secret key in the in the states, and then we clock the state until we reach to initial state. So, this is the initialization state wants to reach to that are initial state then we start the generating the keystream by again clocking. So, depending on the size of the plaintext, we need to get this many keystream. So, this is the general structure.

So, here also we have we have this key initialization state. So, we have this IV. So, here IV key is 80 bits, which is the secret key shared between Alice and Bob are 80 bits. So, this is totally stored into the NLFSR. The key we are storing into the NLFSR because this NLFSR is also 80 bit. And the IV is basically 64 bits and remaining. So, this is LFSR is also 80 bits and remaining 60 bits we are just putting constant like 1 1 1 1. So, 16 plus 64 bits IV, so this is 80 bit. So, we are putting these secret keys 80 bits filling the NLFSR and 64 bit is the IV side. So, we are putting this for 64 bit of the LFSR and remaining 16 bit we are putting 1 1 1 1 like this. And then we run this for 160 round we clock this to stream for 160 round. And after that, we reach to the initial state we run this for 160 round then we reach to the initial state. Then after that we do the clocking to get the key stream, so that is the idea.

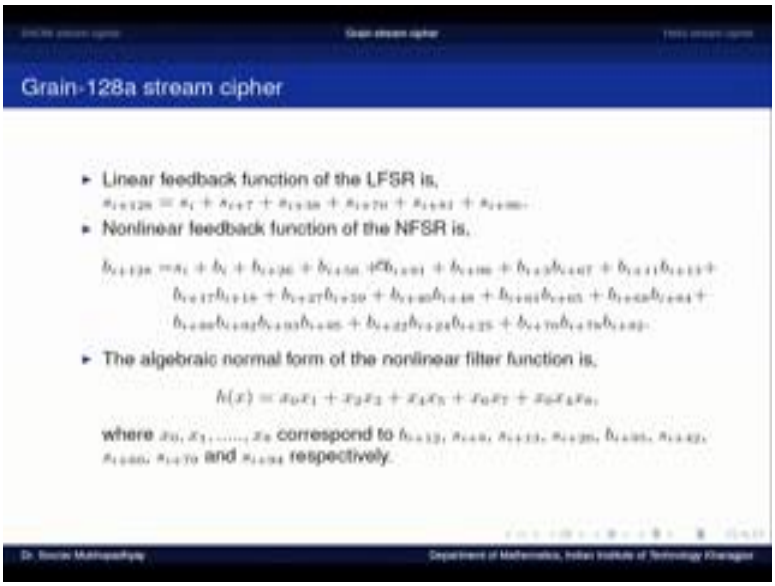
So, we just initialize this key to this NLFSR and we take 64 bit IV to this first 64 bit and remaining 16 bit we put one and this we clock this system for 160 times and then we reach to the initial state of the LFSR and NLFSR. Then we start clocking this to generate the keystream. So, this is the key generation step.

(Refer Slide Time: 22:23)



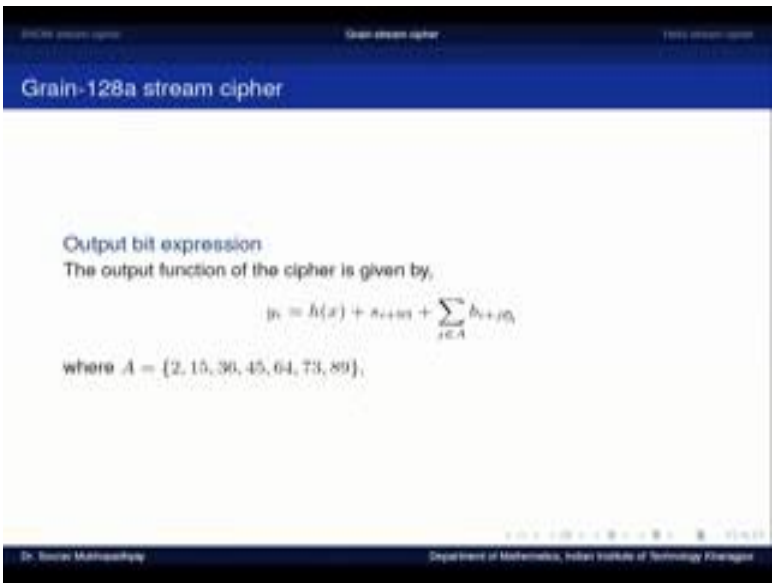
And now this is another version of the grain, which is called grain 128 a. So, this is the authentication purpose. So, use some bits for the authentication purpose. And here what we are doing, so this as similar structures. So, only thing here we are taking the 7th bit. So, this is basically 6th bit of this and we are having the f function this f function is given over here. So, this is the feedback function.

(Refer Slide Time: 22:54)



So, this is the state updation function, this is the state updation function of the NLFSR you have g function we are taking this output you are XORing with this. And this is basically you are updating the NLFSR and this h you are taking 7th bit of the LFSR and 2 bit from the NLFSR. And then we apply this rule and then this is the 7th bit and we are XORing this. So, this is the function like this. So, we have this x 2, x 2 these are the bit, and this is our h function, and this is the state update function for this LFSR and this is the state update function for NLFSR.

(Refer Slide Time: 23:53)



And the output will be like this; this is the output that the key initialization state is here. Here, what we are doing in key initialization state. So, here also we have 80 bit and we have 80 bit NLFSR, 80 bit LFSR. So, here also secret key is 80 bit. So, we put this 80 bit into NLFSR and sorry this is 128 bits. So, this NLFSR is 128 bit this is also 128 bit. So, our key is 128 bit at which we put it into the NLFSR and the IV is 96 bit. So, we put this first 96 bit of these are the LFSR, IV in the first 96 bit in the LFSR and that followed by all the ones and last one is 0. So, that is the initial storing of this n LFSR and the LFSR bit.

And then we run this for 256 times, we do this clocking for 256 times and then we start generating the key stream bit, but we just do the first 64 bit key stream, we will use for the authentication purpose. So, after this 256 bits, after this 256 clocking, we reach to the initial state the value of the LFSR and NLFSR are called the initial state then we can start generating the key streams now we start generating the keystream by clocking again, but now this is also authentication is required. So, we run this for 64 bits; and this 64 bits, we use for the authentication purpose. First 64 bits is stream, and then remaining after that we will just start generating the key stream bit. So, this is the idea.

(Refer Slide Time: 26:09)

So, in here we discussed this. Now, we talk about stream cipher which is asynchronized stream cipher. We have seen the synchronized stream cipher where this keystream are basically coming from the keystream of function of the state basically, but here

keystream are not function of the plaintext or ciphertext. But here in the modern stream cipher they are involving keystream to be a function of the feedback of the plaintext ciphertext to avoid this algebraic attack or may some other correlation attack of bit. So, this is asynchronized stream cipher where we involve this keystream even in the state updation function also we remember the plaintext.

So, one example of this is helix. So, this is the stream cipher. So, this is the state and this is the next update state. So, here if you see, we have used the plaintext. So, this is basically a 32 bit. So, this is basically beta is XORing. This is the rotation operation cycle rotation operation and these are the basically keys. So, these involve the plaintext, so that is why it is called asynchronized stream cipher. So, this will start one state and will get the next state. So, this is one example which where the state update function and the keystream generation is having involved plaintext bit.

(Refer Slide Time: 27:53)

So that that will avoid the known of known attacks, but that is design is little complicated. So, this is called asynchronized stream cipher.

Thank you.