

**Secure Computation - Part I**  
**Prof. Ashish Choudhury**  
**Department of Computer Science**  
**International Institute of Information Technology, Bangalore**

**Module - 1**  
**Lecture - 8**  
**Additive Secret Sharing**

(Refer Slide Time: 00:32)

## Lecture Overview

- Additive Secret-Sharing
  - ❖ Construction
  - ❖ Analysis

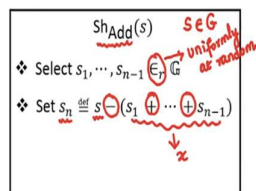


Hello everyone. Welcome to this lecture. So, the plan for this lecture is as follows: We will see a special instantiation of secret-sharing which we call as additive secret-sharing, and we will see its security analysis.

(Refer Slide Time: 00:47)

## $t=n-1$ $(n-1)$ -out-of- $n$ Secret-Sharing Scheme

- **Access Structure:** the entire set of  $n$  parties
- **Forbidden Structure:** any subset of  $n-1$  or a smaller number of parties }  $t = n-1$
- Operations performed over a group  $(G, \oplus)$ 
  - ❖ Secrets and shares are elements of  $G$
- Intuition: secret  $s$  divided into  $n$  **random shares**, such that:
  - ❖ Sum of the shares is  $s$
  - ❖ Any subset of  $n-1$  shares is independent of  $s$



So, this additive secret-sharing is also called as  $n - 1$  out of  $n$  secret-sharing scheme. Namely, it is a special case of threshold secret-sharing, where  $t = n - 1$ . Namely, here you require to have a sharing mechanism in such a way that only when all the  $n$  shareholders make their shares available, then the secret can be reconstructed back. Namely, your access structure consists of just 1 subset, namely the entire subset of  $n$  parties.

And hence, your forbidden structure or unauthorised subsets will consist of  $n - 1$  or less number of parties. That means, we want a mechanism, if any  $n - 1$  or less number of parties pool their shares, they should fail to learn the secret  $s$ . Namely, the probability distribution of their shares should be independent of the underlying secret. So, let us see the instantiation of this  $n - 1$  out of  $n$  secret-sharing.

And here, we will perform all the operations, namely the sharing, reconstruction, all the operations we will perform over an abstract group  $\mathbb{G}$  with an abstract  $+$  operation. Remember, this is an abstract  $+$  operation that need not be the numeric or the integer or a real number addition. And here, the secrets and the shares will be elements of the group. That means, the dealer's secret will be an element of the group, the shares also will be element of the group and so on.

So, the main intuition behind this sharing algorithm is the following: The sharing algorithm basically divides the secret into  $n$  random shares, random looking shares such that the sum of the shares; and by sum, I need not mean in the integer sum sense; by sum, I mean as per the group operation. The sum of the shares, the  $n$  random shares should be equal to the secret  $s$ . But, out of those  $n$  shares, if I focus on any  $n - 1$  shares, that should be independent of the secret  $s$ .

That is a rough intuition behind this secret-sharing scheme. So, now, let us go directly to the sharing algorithm. So, I denote the sharing algorithm of this scheme as a  $Sh_{Add}$ . Add denotes the additive secret-sharing. The input is the secret  $s$ , which is going to be an element of the group. Now remember, the sharing algorithm has to be randomised. Let us see where exactly the internal randomness is coming into picture.

So, the first step of the algorithm will be the following: It will randomly pick  $n - 1$  shares, namely the first  $n - 1$  shares. Well, it could be the any  $n - 1$  shares, but just for the sake of simplicity and without loss of generality, assume that the sharing algorithm picks the first  $n - 1$  shares uniformly at random. From where? From the group itself. That means, remember I said, the shares are going to be elements of the group; what I am saying is that, pick the  $n - 1$  shares as random group elements.

This notation,  $\epsilon_r$  belongs and then subscript  $r$ , denotes uniformly at random. So, it could be the case that all  $s_1$  to  $s_{n-1}$  are the same value, because you are picking  $s_1$  independently,  $s_2$  independently,  $s_{n-1}$  independently. So, there is a non-zero probability that all of them are the same group element, or, they could be all distinct. That is also a non-zero probability event. Or it could be the case that all the odd indexed shares are the same or the even indexed shares are the same and so on.

So, all this events can occur with non-zero probability, because you are picking the first  $n - 1$  shares for the secret  $s$ , uniformly at random from your underlying group. Now, we have to define or compute the value of the  $n$ th share. The  $n$ th share is computed as follows. You add the first  $n - 1$  shares. And by adding I mean, as per the underlying group addition operation. You compute the additive inverse of  $s_1 + s_2 + \dots + s_{n-1}$  and add it with your secret  $s$ .

Because, again, the interpretation of this minus operation is depending upon your underlying group. Here it means, basically we are adding the additive inverse. So, let us denote the summation of first  $n - 1$  shares as  $x$ .  $s - x$  basically means I am adding the additive inverse of  $x$  to the secret  $s$ . Or, in other words, the way to interpret is, that my  $s_n$  should be such that the summation of  $x$  and  $s_n$  should give me the secret  $s$ . That is the way the  $n$ th share is computed.

**(Refer Slide Time: 06:43)**

## $t=n-1$ $(n-1)$ -out-of- $n$ Secret-Sharing Scheme

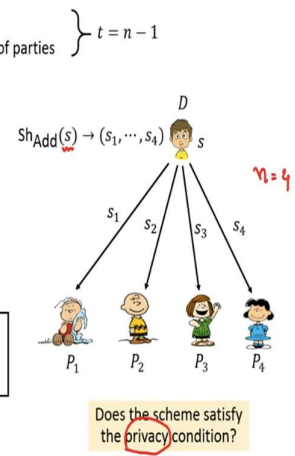
- Access Structure: the entire set of  $n$  parties
- Forbidden Structure: any subset of  $n-1$  or a smaller number of parties }  $t = n-1$
- Operations performed over a group  $(\mathbb{G}, \oplus)$ 
  - ❖ Secrets and shares are elements of  $\mathbb{G}$
- Intuition: secret  $s$  divided into  $n$  random shares, such that:
  - ❖ Sum of the shares is  $s$
  - ❖ Any subset of  $n-1$  shares is independent of  $s$

$Sh_{Add}(s)$

- ❖ Select  $s_1, \dots, s_{n-1} \in_r \mathbb{G}$
- ❖ Set  $s_n \triangleq s - (s_1 + \dots + s_{n-1})$
- ❖ Output  $s_1, \dots, s_n$

$Rec_{Add}(s_1, \dots, s_n)$

- ❖ Output  $s_1 + \dots + s_n$



Remember, all these actions are done as part of your sharing algorithm. And now, the algorithm outputs, the shares  $s_1$  to  $s_n$ . And if dealer wants to share the secret  $s$ , it will run this sharing algorithm; compute  $s_1$  to  $s_n$ ; and the value of  $s_i$  will be given to the party number  $P_i$ , over the private channel, between the dealer and the party number  $P_i$ . So, now, you can see that why this algorithm is a randomised algorithm.

Say for instance, if you execute this sharing algorithm with the same input  $s$  twice in succession, will you get the same values of  $s_1$  to  $s_n$ ? No, need not be. Because, the first time you run the sharing algorithm with input  $s$ , the shares  $s_1$  to  $s_{n-1}$  might be different. In fact, they are independent compared to the values of  $s_1$  to  $s_{n-1}$  when you run the sharing algorithm for the secret  $s$  in the second attempt.

And that is why this sharing algorithm is randomised. So, again, let us take the case of  $n = 4$ . If dealer wants to share the secret  $s$  as per the additive secret-sharing, in such a way that, even if up to 3 parties combined their shares, they should fail to learn the secret. The dealer will do the following: It will compute the shares  $s_1, s_2, s_3, s_4$ , as per the secret  $s$ , using this sharing algorithm.

And the respective shares, it will communicate privately over the channel, between the dealer and the  $i$ th party. Now, what will be the reconstruction algorithm? And for reconstruction algorithm, we need a mechanism where, if all the  $n$  shareholders make their shares available, we should get back the secret  $s$ . So, the input for the reconstruction algorithm will be the vector of  $n$  shares  $s_1$  to  $s_n$ . And it is very simple to get back the secret  $s$ .

What you have to do? You just add the  $n$  shares  $s_1$  to  $s_n$ . And because of the fact that your summation of  $s_1$  to  $s_n$  is the secret  $s$ , you will get back the correct output. That is a very simple sharing algorithm and the reconstruction algorithm, where the operations are performed over a group. So, correctness condition is trivial to verify, because we have only 1 authorised subset, namely the entire set of shareholders.

And if the entire set of shareholders come together by running the *Rec* algorithm, of course, they can construct back the value of your secret  $s$ . But what about privacy condition? Does this scheme achieves the privacy condition? And for privacy condition, remember, we have to formally argue that any subset of  $n - 1$  or less number of shareholders, the probability distribution of their shares is independent of the value of the secret which dealer has shared.

That is what we have to formally argue. And intuitively, before going into the proof, let me give you an intuition that why the privacy condition is true. We need all the  $n$  pieces to get back the secret. Even if 1 piece is missing, that 1 piece could have been any value from the group. And depending upon what value from the group would have been that missing piece, implies a corresponding secret.

That is why, from the viewpoint of  $n - 1$  shareholders who are not supposed to get back the secret, it could be any value of the missing piece, as the missing share. And hence, it could be any value from the group, which dealer could have shared.

**(Refer Slide Time: 10:51)**

### (n - 1)-out-of-n SS: Instantiation

□ Operations performed over the group  $(\{0, 1\}^l, \oplus, \otimes)$


$\oplus$  "+"  
 $\otimes$  "XOR"

$a \oplus b = 0$  if  $a = b$   
 $= 1$  if  $a \neq b$

$a = a_0 a_1 \dots a_{l-1}$   
 $b = b_0 b_1 \dots b_{l-1}$

---

$a \otimes b = (a_0 \otimes b_0) (a_1 \otimes b_1) (a_2 \otimes b_2) \dots$



So, I will make this intuition more specific by taking concrete examples of the groups. So, I will take 2 specific groups, where I will show you the instantiation of this additive secret-sharing. And for both these groups, we will analyse the privacy property. So, let us take the first candidate group, where my  $\mathbb{G}$ , group  $\mathbb{G}$  consists of all bit strings of length  $l$  bits. And my abstract plus operation is the bitwise XOR operation  $\oplus$ .

So, I am sure all of you know what is the XOR operation. If you have 2 bits  $a$  and  $b$ , and if you perform their XOR, then the result will be 0 if  $a = b$ ; otherwise, it will be 1. In some sense, you can imagine this XOR operation to be addition modulo 2. Now, what is this bitwise XOR operation? So, under the subscript, you have this  $l$ . So, since my elements are now going to be  $l$  bit long binary strings, if I am given 2 long binary strings  $a$  and  $b$  consisting of  $l$  bits  $a_0, a_1$  up to  $a_{l-1}$ ; and in the same way,  $b$  is  $b_0, b_1$  up to  $b_{l-1}$ , then  $a \oplus_l b$  bitwise is basically the result of  $a_0 \oplus b_0$ .

That will be the first bit or LSB. That is a first bit, not the LSB. The next bit will be  $a_1 \oplus b_1$ . The next bit will be  $a_2 \oplus b_2$  and so on. That is a definition of this bitwise XOR operation. And it is easy to see that this indeed is a group, because all your group properties are satisfied. You take the closure property; you take any  $l$  bit long string; perform the bitwise XOR operation; the result again will be an  $l$  bit long string. So, the closure property is satisfied. XOR satisfies the associative property.

**(Refer Slide Time: 13:25)**

**$(n - 1)$ -out-of- $n$  SS: Instantiation**

□ Operations performed over the group  $(\{0, 1\}^l, \oplus_l)$

$D$   
 $s \in \{0, 1\}^l$

$P_1$     $P_2$     $P_3$     $P_4$

$0 \rightarrow 0000 \dots 0$   
 $x \rightarrow (x^{-1}) \text{ is } 0 \oplus (x_0 \dots x_{l-1})$   
 $= x_0 \oplus x_1 \dots x_{l-1}$

The binary string all 0, of length  $l$  bits; this is your additive identity element. And if you have  $x$ , then corresponding to; where  $x$  is actually a bit string. Then  $-x$  or  $x$  inverse is, which is

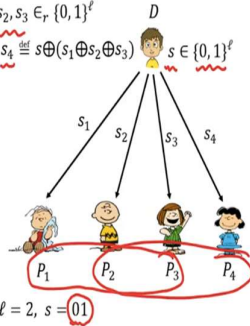
same as the string  $x$  itself. Namely,  $x$  inverse is  $x$  itself, because, if you perform the XOR of  $x$  and  $x$ , both of them are  $l$  bit long string. You will get back the inverse additive identity element, namely the 0. So, it satisfies; in fact, this is an abelian group because the XOR operation even satisfies your commutative property.

**(Refer Slide Time: 14:42)**

## (n - 1)-out-of-n SS: Instantiation

Operations performed over the group  $((0, 1)^l, \oplus_l)$

- ❖  $s_1, s_2, s_3 \in_r \{0, 1\}^l$
- ❖ Set  $s_4 \stackrel{\text{def}}{=} s \oplus (s_1 \oplus s_2 \oplus s_3)$   $s \in \{0, 1\}^l$



So, now, imagine dealer has a secret, which is some  $l$  bit long string. And if it wants to secret share it, what it will do is, it will first pick 3 random strings of length  $l$  bits uniformly at random. And they will be the first 3 shares, namely shares of  $P_1, P_2, P_3$ . And the fourth share will be  $s$  minus the summation of the first 3 shares. And remember, summation here is basically the XOR operation. And subtraction here is also basically the XOR operation.

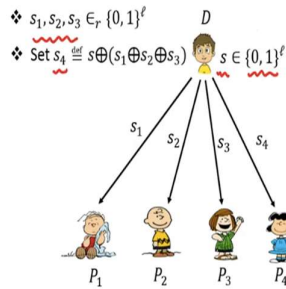
So, the fourth share is such that the XOR of shares  $s_1, s_2, s_3$  and  $s_4$  gives you back together the secret  $s$ . And now, the dealer will distribute the shares  $s_1, s_2, s_3, s_4$  to the respective parties. So, now, let us take a concrete value of  $l$  and see the real execution or one of the executions possible. So, imagine  $l = 2$ . And the secret which dealer wants to share is the binary string 01, which only dealer knows.

And remember, our goal here is to ensure that once the binary string 01 is shared; now, then any set of this, any subset of 3 parties among these 4 parties, if they are curious and try to find out what is the underlying secret the dealer has shared, based on what shares they have received, they will fail completely. That is what we have to ensure.

**(Refer Slide Time: 16:23)**

## (n - 1)-out-of-n SS: Instantiation

Operations performed over the group  $((0, 1)^\ell, \oplus)$



Example:  $\ell = 2, s = 01$

$\clubsuit$  Let  $s_1 = 10, s_2 = 00, s_3 = 01$   
 $\clubsuit$  Then  $s_4 = 10 \quad (10 \oplus 00) \oplus (01 \oplus 00) = 01$



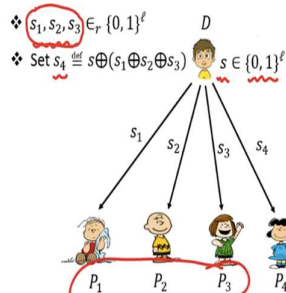
So, imagine that, in the first step, the dealer picks the share  $s_1$  as the string 10, the share  $s_2$  as the string 00, and the share  $s_3$  as the string 01. Well, remember, it could be the case that  $s_1$  takes any possible binary string of length 2. All, any binary string of length 2 can be a candidate  $s_1$ , because  $s_1$  is picked uniformly at random. In the same way,  $s_2$  can take the value as any binary string of length 2. And  $s_3$  can take the value, any binary string of length 2, because all of them are picked uniformly at random.

Now, based on the value of  $s_1, s_2$  and  $s_3$ , the  $s_4$  will be 10, the binary string. Because, now, if you see that if you perform the XOR of 10 with 00, with 01, with 10, you will get the value 01 which is your secret  $s$ . But it is not the case that these are the only possible values of  $s_1, s_2, s_3, s_4$ , for the secret  $s = 01$ .

**(Refer Slide Time: 17:41)**

## (n - 1)-out-of-n SS: Instantiation

Operations performed over the group  $((0, 1)^\ell, \oplus)$



Example:  $\ell = 2, s = 01$

$\clubsuit$  Let  $s_1 = 10, s_2 = 00, s_3 = 01$   
 $\clubsuit$  Then  $s_4 = 10$

Consider the unauthorized subset  $B = \{P_1, P_2, P_3\}$

$s_1, s_2, s_3$  picked independently of  $s \Rightarrow \{g_B(00)\} \equiv \{g_B(01)\}$   
 $\equiv \{g_B(10)\} \equiv \{g_B(11)\}$





It could be other possible vectors of  $s_1, s_2, s_3, s_4$  as well, which could be the potential shares for the secret  $s$ , if the dealer runs this sharing algorithm again for secret-sharing, the secret 01. So, now, let us try to understand the privacy property, whether the secret-sharing scheme satisfies the privacy requirement or not. And imagine that I consider the unauthorised subset consisting of parties 1, 2 and 3.

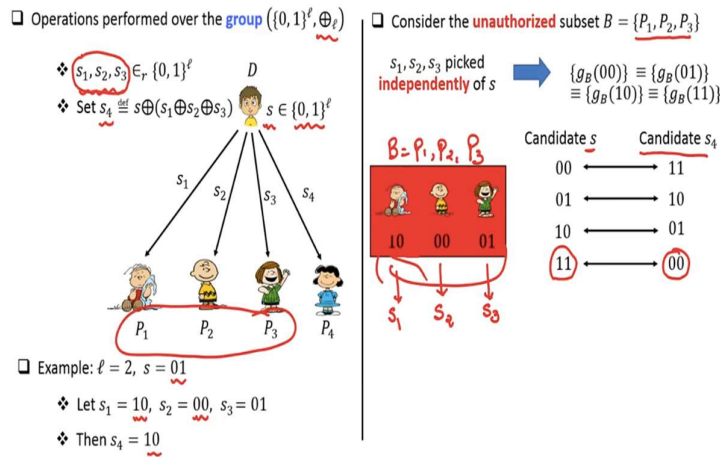
Can we say that based on the values of  $s_1, s_2, s_3$ , they learn anything about the underlying secret  $s$ ? Can they say that, okay, the secret  $s$  is 00 or 01 or 10 or 11, based on the values of  $s_1, s_2, s_3$ ? And the answer is no. Because, if you see the steps of the algorithm,  $s_1, s_2, s_3$ , they have absolutely no relationship with the secret  $s$ . They are picked independently of what is the value of the secret  $s$ .

$s$  could be 00,  $s$  could be 01,  $s$  could be 10,  $s$  could be 11;  $s_1, s_2, s_3$  have got absolutely nothing to do with the value of the secret. And that is why we can say that the random variable  $g_B$ ; and remember from the earlier lecture,  $g_B(s)$  denotes the random variable; the  $g_B$  is the random variable denoting the value of the shares  $s_1, s_2, s_3$ , because, I am taking  $B$  to be  $P_1, P_2, P_3$ .

$g_B(00)$  means, the value of the shares  $s_1, s_2, s_3$ , if 00 would have been shared by the dealer.  $g_B(01)$  means, the value of the shares  $s_1, s_2, s_3$ , if 01 is the secret.  $g_B(10)$  means, the value of  $s_1, s_2, s_3$ , if 10 is the shared secret. And  $g_B(11)$  means, the value of the shares  $s_1, s_2, s_3$ , if 11 is the secret. And this variables  $g_B$ , it is a random variable; it is not a variable which always takes a fixed value.

That is why, we are now going to talk about the probability distribution. Remember,  $\{g_B(00)\}$  means, the probability that  $s_1, s_2, s_3$  takes different values with different probability, given that 00 is the secret which is shared and so on. Similarly, you can interpret the remaining variables.  
**(Refer Slide Time: 20:31)**

## (n - 1)-out-of-n SS: Instantiation



So, since we have fixed my  $B$  to be  $P_1, P_2, P_3$ , in this specific case, the shares that  $P_1, P_2, P_3$  have learnt are 10, 00, 01. So, suppose these 3 people, they call each other and then they say to each other, okay, these are the shares that we have received from the dealer. And now they are trying to learn the dealer's secret. We have to ensure that, actually, we have to show that they will fail to do that.

So, by calling each other or whatever mechanism if they come and meet and then they tell each other that, okay, this is the share that they have received individually from the dealer, they will come to the fact that the shares that dealer has given to them is 10, 00, 01 respectively. And now they are making an hypothesis in their mind. They are trying to compute that what could be the probability that dealer has actually shared the secret 00, and the shares learned by them are these shares, or, what is the probability that dealer has shared the secret 01, and they have seen these shares and so on.

So, from the viewpoint of the parties in  $B$ , the secret which dealer could have shared can take 4 candidate values, because, remember, the value of  $l$  is publicly known. The description of the secret space is publicly known. So, these 3 parties  $P_1, P_2, P_3$  know that the candidate  $s$  could be 00, 01, 10 or 11. Now, it could be the case that dealer has actually shared the secret 00 and  $s_1$  is 10,  $s_2$  is 00,  $s_3$  is 01 and  $s_4$  is 11.

It is quite possible. Because, if indeed you take the XOR of 10, 00, 01 and 11, you get the secret to be 00. But the parties in  $B$  do not know the value of  $s_4$ , because  $s_4$  is right now not in this collection  $B$ . Only when  $s_4$ ; if of course  $s_4$  come with them, then together they are authorised;

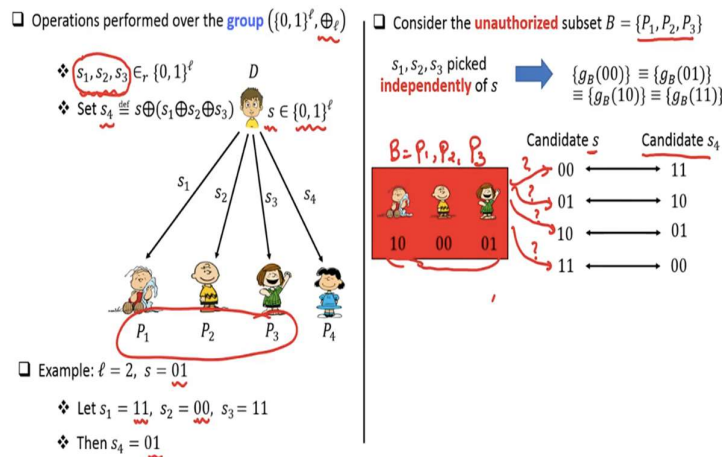
they constitute an authorised set. We are right now analysing the case when an unauthorised collection of parties are trying to learn the secret.

So, if the parties in  $B$  make an hypothesis that the dealer has shared the secret  $s$  being 00, that is quite possible, because, it could be the case that the value of  $s_4$  is 11. On the other hand, if the parties in  $B$  make the hypothesis that the candidate  $s$ , the secret  $s$  which the dealer has shared is 01, that is again quite possible, because, if the missing  $s_4$  is 10 and if  $s_1$  is 10,  $s_2$  is 00,  $s_3$  is 01, then, indeed the secret that dealer has shared is 01.

In the same way, if the parties in  $B$  make the hypothesis that, okay, the shares that they have seen correspond to the secret  $s$ , that is quite possible, because, it could be the case that the missing  $s_4$  which the parties in  $P_1, P_2, P_3$  do not have is 01. And in the same way, if the parties in  $B$  make the hypothesis that dealer has shared the secret 11, that is quite possible because, it could be the case that the missing  $s_4$  is 00. So, what we have argued here is the following:

**(Refer Slide Time: 24:49)**

### (n - 1)-out-of-n SS: Instantiation



Based on the values of  $s_1, s_2, s_3$ , that the parties in  $B$  learn, they cannot pinpoint whether the dealer has actually executed the sharing algorithm as per the first configuration here or the second configuration or the third configuration or the fourth configuration, because each of these 4 configurations are equiprobable with same probability. And that is why, learning the shares 10, 00, 01, does not allow or help the parties in  $B$  to learn anything about the underlying secret  $s$ . Now, assume that the dealer, the same dealer, again runs the secret-sharing algorithm. And assume that this time also, his value of secret remains the same.

**(Refer Slide Time: 25:46)**

## (n - 1)-out-of-n SS: Instantiation

Operations performed over the group  $(\{0, 1\}^\ell, \oplus)$

- ❖  $s_1, s_2, s_3 \in_r \{0, 1\}^\ell$
- ❖ Set  $s_4 \equiv s \oplus (s_1 \oplus s_2 \oplus s_3)$

Consider the unauthorized subset  $B = \{P_1, P_2, P_3\}$

$s_1, s_2, s_3$  picked independently of  $s$   $\Rightarrow$   $\{g_B(00)\} \equiv \{g_B(01)\}$   
 $\equiv \{g_B(10)\} \equiv \{g_B(11)\}$

Candidate $s$	Candidate $s_4$
00	11
01	10
10	01
11	00

Candidate $s$	Candidate $s_4$
00	00
01	01
10	10
11	11

Example:  $\ell = 2, s = 01$

- ❖ Let  $s_1 = 10, s_2 = 00, s_3 = 01$
- ❖ Then  $s_4 = 10$

Of course, the parties  $P_1, P_2, P_3, P_4$  are not aware of this fact. It is only the dealer who again wants to share the same secret. And now, suppose, during the first step, when the dealer is picking the shares  $s_1, s_2, s_3$  randomly, the values  $s_1, s_2, s_3$  takes these values. The shares  $s_1, s_2, s_3$  takes these values, 11, 00, 11. It could be possible, right? If that is the case, then the share  $s_4$  will be 01.

And now, you can see, earlier, my  $s_1$  was something else; it was 10. But now, my  $s_1$  is 11. Earlier, my  $s_2$  was 00, for the same secret. Now, my  $s_2$  is 00. Fine. Earlier, my  $s_3$  was 01. Now, my  $s_3$  is 11. And based on what were the earlier values of  $s_1, s_2, s_3$ , my  $s_4$  was 10. But now, my  $s_4$  is 01, but my secret is the same secret. That means, for the same secret 01, the vector of 4 shares could take different values with different probabilities.

**(Refer Slide Time: 26:54)**

## (n - 1)-out-of-n SS: Instantiation

Operations performed over the group  $(\{0, 1\}^\ell, \oplus)$

- ❖  $s_1, s_2, s_3 \in_r \{0, 1\}^\ell$
- ❖ Set  $s_4 \equiv s \oplus (s_1 \oplus s_2 \oplus s_3)$

Consider the unauthorized subset  $B = \{P_1, P_2, P_3\}$

$s_1, s_2, s_3$  picked independently of  $s$   $\Rightarrow$   $\{g_B(00)\} \equiv \{g_B(01)\}$   
 $\equiv \{g_B(10)\} \equiv \{g_B(11)\}$

Candidate $s$	Candidate $s_4$
00	11
01	10
10	01
11	00

Candidate $s$	Candidate $s_4$
00	00
01	01
10	10
11	11

Example:  $\ell = 2, s = 01$

- ❖ Let  $s_1 = 11, s_2 = 00, s_3 = 11$
- ❖ Then  $s_4 = 01$

So, now imagine that a same unauthorised subset of parties  $P_1, P_2, P_3$  come together. And again they call each other and say, okay, this time we got the shares 11, 00, 11 from the dealer. Can they say anything whether the candidate  $s$  which dealer has shared is 00, 01, 10, 11, based on their shares? And again, the candidate  $s$  for them, this time also could be 00, 01, 10 or 11.

And it turns out that each of this candidate configuration could be possible for them. They cannot pinpoint whether dealer has actually run the secret-sharing algorithm with candidate  $s$  being 00 or candidate  $s$  being 01 or  $s$  being 10 or  $s$  being 11. Because, for each candidate  $s$  that these people in  $B$  think in their mind, there is a corresponding unique  $s_4$  which is missing for them.

And hence, it is equiprobable that the shares 11, 00, 11, could be the shares for the secret  $s$  being 00, or could be the shares for the secret  $s$  being 01, or they could be the shares corresponding to the secret  $s$  being 10, or they could be the shares corresponding to the secret being 11 with equal probability. Which shows that the random variable  $g_B$  with respect to  $B$  being  $P_1, P_2, P_3$ , is uniformly distributed. It is independent of whether the secret is 00, 01, 10 or 11.

**(Refer Slide Time: 28:38)**

### $(n - 1)$ -out-of- $n$ SS: Instantiation

Operations performed over the group  $(\{0, 1\}^\ell, \oplus_\ell)$

- ❖  $s_1, s_2, s_3 \in_r \{0, 1\}^\ell$
- ❖ Set  $s_4 \stackrel{\text{def}}{=} s \oplus (s_1 \oplus s_2 \oplus s_3)$

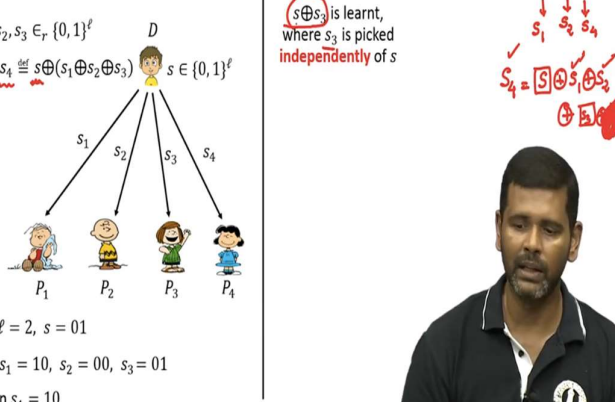
Example:  $\ell = 2, s = 01$

- ❖ Let  $s_1 = 10, s_2 = 00, s_3 = 01$
- ❖ Then  $s_4 = 10$

Consider the **unauthorized** subset  $B = \{P_1, P_2, P_3\}$

$s \oplus s_3$  is learnt, where  $s_3$  is picked independently of  $s$

$s_4 = s \oplus s_1 \oplus s_2 \oplus s_3$



So, that is an analogy, that is an analysis when  $P_1, P_2, P_3$  are corrupt. Now, you might argue that, what if my unauthorised subset consists of party  $P_4$ ? Because, as per the sharing algorithm, the fourth share or the last share has actually something to do with the secret  $s$ . If the first  $n - 1$  shareholders are corrupt; corrupt in the sense, they constitute an unauthorised subset; of course,

they does not learn anything about the underlying secret, because the probability distribution of their shares is completely independent of the secret.

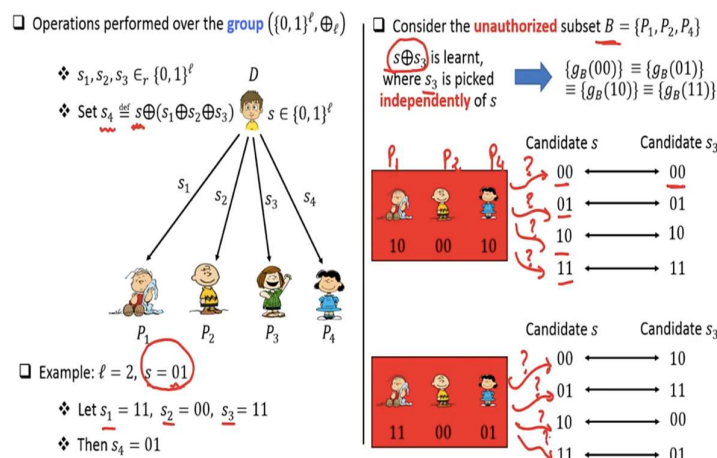
But what if in the unauthorised collection, nth party is present? Because, the share of the nth party is actually depending on your secret  $s$ , as per the formula that we have here in the sharing algorithm. It turns out that if I consider an unauthorised subset consisting of  $n - 1$  parties, say for instance, party number 1, 2, 4; then, if I take party 2, it has the share  $s_2$ ; if I take the party 4, it has the share  $s_4$ ; if I take the party  $P_1$ , it has the share  $s_1$ .

And it is known that  $s_4$  is basically the unknown  $s$ . So, let me put the unknown  $s$  in question, this box. That means, its value is not yet known. XOR  $s_1$ ; XOR  $s_2$ ; XOR an unknown  $s_3$ , which is not yet known; XOR  $s_4$ . So, what are the things which are known to this unauthorised subset? So, they know  $s_1$ ; they know  $s_2$ ; they know  $s_4$ . Sorry, not  $s_3$ . So, if they substitute the value of  $s_4$ ,  $s_1$  and  $s_2$ , they come to learn the value of  $s \oplus s_3$ .

Now, does this reveal anything about the secret  $s$ ? Absolutely no, because, the missing piece  $s_3$  here is picked independently at random, uniformly at random by the dealer, independent of the shares  $s_1, s_2, \dots, s_{n-1}$ . And hence, it could be any  $s_3$ , XOR with any  $s$ , which actually the unauthorised parties in  $B$  would have learnt. So, again let us make it more concrete.

**(Refer Slide Time: 31:18)**

### $(n - 1)$ -out-of- $n$ SS: Instantiation



Let us take the example where, say  $l = 2$ ; the secret shared being 01; and these are the shares; and I am now taking an unauthorised collection consisting of 1, 2 and 4. So, party number 1,

party number 2 and party number 4; these are the shares. For them, the candidate  $s$  could be these 4 values. And again, it turns out that for every candidate  $s$  that they think in their mind, that dealer has and with which dealer has run the sharing algorithm, there is a corresponding missing  $s_3$ , which the parties in  $B$  do not learn.

And since  $s_3$  is picked uniformly at random, it could be the case that any of these 4 executions has happened. And the parties in 1, 2 and 4, have seen the shares 10, 00, 10. In the same way, again, if the dealer reruns his sharing algorithm with the same secret, then, depending upon the values of  $s_1, s_2, s_3$  here,  $s_4$  could take this value. And now, let us take the view of the same parties in  $B$ .

This time, their shares are now different. But again, they cannot pinpoint whether this time they have actually seen the shares corresponding to dealer's secret being 00, 01, 10, 11, because, again, this time also, it could be any of these 4 configurations.

**(Refer Slide Time: 32:58)**

**$(n - 1)$ -out-of- $n$  SS: Instantiation**

□ Operations performed over the group  $(\mathbb{Z}_m, +_m)$       $\mathbb{Z}_m = \{0, \dots, m-1\}$

❖  $s_1, s_2, s_3 \in_r \mathbb{Z}_m$

❖  $s_4 \equiv (s - (s_1 + s_2 + s_3)) \pmod m$

Now, let us take another candidate group and see the working of the additive secret-sharing. Now, this time, my group is the group  $\mathbb{Z}_m$ . And remember,  $\mathbb{Z}_m$  consists of the integers 0 to  $m - 1$ . And this is my addition modulo  $m$  operation. The secret should be a member of  $\mathbb{Z}_m$ . To share a secret, dealer will randomly pick the first 3 shares as random elements from the set  $\mathbb{Z}_m$ .

And now, the fourth element or the fourth share should be such that, the summation of  $s_1 + s_2 + s_3 + s_4$  modulo  $m$  should give me the secret  $s$ . That is how the fourth piece should be computed. And now, once computed, dealer will distribute the shares like this.

**(Refer Slide Time: 33:59)**

### (n - 1)-out-of-n SS: Instantiation

□ Operations performed over the group  $(\mathbb{Z}_m, +_m)$

- ❖  $s_1, s_2, s_3 \in_r \mathbb{Z}_m$
- ❖  $s_4 \equiv (s - (s_1 + s_2 + s_3)) \pmod m$

□ Consider the unauthorized subset  $B = \{P_1, P_2, P_3\}$

$s_1, s_2, s_3$  picked independently of  $s \rightarrow \{g_B(0)\} \equiv \{g_B(1)\} \equiv \{g_B(2)\} \equiv \{g_B(3)\}$

□ Example:  $m = 4, s = 3$

- ❖ Let  $s_1 = 1, s_2 = 1, s_3 = 3$
- ❖ Then  $s_4 = 2$

Candidate $s$	Candidate $s_4$
0	3
1	0
2	1
3	2

So, again, let us take an example and try to analyse whether the privacy property is achieved. Imagine that the parties agree upon the modulus being  $m = 4$ . Remember, the value of the modulus, underlying group operations, all these details are publicly known. And say the dealer's secret is  $s = 3$ . It has to pick the first 3 shares uniformly at random. Suppose, it picks the shares 1, 1 and 3.

So, now you see, 1, this first share and the second share turns out to be same. It is quite possible, because they are picked uniformly at random and independent of each other. Now, based on the first 3 shares, the last share turns out to be 2. Because, if indeed you add 2, 1, 1 and 3, and then take modulo 4, you get back the result to be 3, which is your secret  $s = 3$ . So, now let us consider an unauthorised subset 1, 2 and 3.

Again, the shares  $s_1, s_2, s_3$ , it has got nothing to do with the secret  $s$ . And hence, the probability distribution of the shares that the parties 1, 2 and 3 see during the execution of the protocol should be independent of the actual secret. So, in this specific case, what are the shares they are saying? So, this is 1 candidate value of  $g_B$ . The shares  $s_1, s_2, s_3$  could take the values 1, 1 and 3.



And these shares could be the shares for the secret 0, if the fourth share would have been 3. The same 3 shares could be the shares for the secret 1, if the fourth share would have been 0. The same 3 shares could be the shares for the secret 2, if the fourth share would have been 1. And the same 3 shares could be the shares for the secret 3, if the fourth share would have been 2.

But since the fourth share could be any of these 4 values with equal probability, hence it could be any of these configurations that actually the parties in  $B$  have participated. Again, let us assume that dealer reruns the sharing algorithm, but his secret remains the same.

**(Refer Slide Time: 36:19)**

### (n - 1)-out-of-n SS: Instantiation

Operations performed over the group  $(\mathbb{Z}_m, +_m)$

- $s_1, s_2, s_3 \in_r \mathbb{Z}_m$
- $s_4 \equiv (s - (s_1 + s_2 + s_3)) \pmod m$

Example:  $m = 4, s = 3$

- Let  $s_1 = 3, s_2 = 3, s_3 = 3$
- Then  $s_4 = 2$

$\mathbb{Z}_m = \{0, \dots, m-1\}$

Consider the unauthorized subset  $B = \{P_1, P_2, P_3\}$

$s_1, s_2, s_3$  picked independently of  $s \Rightarrow \{g_B(0)\} \equiv \{g_B(1)\} \equiv \{g_B(2)\} \equiv \{g_B(3)\}$

Candidate $s$	Candidate $s_4$
0	3
1	0
2	1
3	2

Candidate $s$	Candidate $s_4$
0	3
1	0
2	1
3	2

And now, this time, it results in a different vector of shares, because, now the values of  $s_1, s_2$  and  $s_3$ , all of them turn out to be the same. Again, the parties will not be knowing this. It is only the dealer who is running this. And based on the values of his first 3 shares, the fourth share turns out to be 2. Now, what will be  $g_B$ ? So, the  $g_B$  will be basically the shares corresponding to the parties in 1, 2, 3, this time.

And this time, their shares are 3, 3, 3. Now, can it be the case that these 3 shares are actually for the secret being 0? Yes, provided the fourth piece would have been 3. Can these shares 3, 3, 3 could be the shares for the secret 1? Yes, provided the share  $s_4$  is 0, and so on. So, again, this time, all candidate  $s$  are possible from the viewpoint of the parties in  $B$ . And hence, the parties in  $B$  does not learn anything.

**(Refer Slide Time: 37:19)**

## (n - 1)-out-of-n SS: Instantiation

□ Operations performed over the group  $(\mathbb{Z}_m, +_m)$

- ❖  $s_1, s_2, s_3 \in_r \mathbb{Z}_m$
- ❖  $s_4 \equiv (s - (s_1 + s_2 + s_3)) \pmod m$

□ Consider the **unauthorized** subset  $B = \{P_2, P_3, P_4\}$

$s - s_1$  is learnt, where  $s_1$  is picked **independently** of  $s$

$\{g_B(0)\} \equiv \{g_B(1)\}$   
 $\equiv \{g_B(2)\} \equiv \{g_B(3)\}$

□ Example:  $m = 4, s = 3$

- ❖ Let  $s_1 = 1, s_2 = 1, s_3 = 3$
- ❖ Then  $s_4 = 2$

Candidate $s$	Candidate $s_1$
0	2
1	3
2	0
3	1

Now, what if fourth party is one of the parties in the unauthorised collection? So, let us take this unauthorised collection where the first party is missing. But the last 3 parties, they call each other and then learn that, okay, their shares are 1, 1 and 2. So, again, similar to the case of where our group was consisting of strings of length 1 bits; if the parties in 2, 3 and 4 combine their shares; combine in the sense, they make their share, the value of their shares available to each other; they will learn something about the secret.

But that something about the secret is basically a difference of  $s$  and  $s_1$ , where  $s_1$  is split independently of the secret  $s$ . Since it could be any  $s_1$ , which the dealer would have picked, learning  $s - s_1$ , the value of, concrete value of  $s - s_1$ , does not help the parties in  $B$  to learn anything about the secret  $s$ , because, it could be any  $s$  and any  $s_1$  whose difference would have been the value which the parties in  $B$  would have learnt.

So, let us make, let us see this concrete. So, again, the parties 2 and 3 and 4 in this specific case, they will be having access to these 3 shares. And they are arguing in their mind together, whether they have actually seen the shares corresponding to secret being 0, secret being 1, secret being 2, secret being 3. Well, again, it could be the case that dealer has shared the secret 0, because the first share was at 2; which together would have resulted in this shares.

Or it could be the case that the share secret was 1, the first piece was 2, and then it resulted in this same shares 1, 3, 2. Or it could be the case that the secret was 2, and the first share was 0, and it resulted in the remaining 3 shares to be 1, 3, 2. And it could be the case that the secret

was 3, and the first share was 1, and then the last 3 shares turned out to be 1, 3, 2. That means, for each candidate  $s$ , there is a corresponding  $s_1$ .

And hence, learning  $s - s_1$  does not allow the parties in  $B$  to infer anything about the secret  $s$ . They are as confused as they were earlier. Now, imagine that the dealer runs the sharing algorithm again with the same secret, but this time, the share vector is something else.

**(Refer Slide Time: 40:03)**

### $(n - 1)$ -out-of- $n$ SS: Instantiation

□ Operations performed over the group  $(\mathbb{Z}_m, +_m)$

- ❖  $s_1, s_2, s_3 \in \mathbb{Z}_m$
- ❖  $s_4 \equiv (s - (s_1 + s_2 + s_3)) \pmod m$

□ Consider the **unauthorized** subset  $B = \{P_2, P_3, P_4\}$

$s - s_1$  is learnt, where  $s_1$  is picked **independently of  $s$**

$\{g_B(0)\} \equiv \{g_B(1)\}$   
 $\equiv \{g_B(2)\} \equiv \{g_B(3)\}$

□ Example:  $m = 4, s = 3$

- ❖ Let  $s_1 = 2, s_2 = 2, s_3 = 2$
- ❖ Then  $s_4 = 1$

Candidate $s$	Candidate $s_1$
0	2
1	3
2	0
3	1

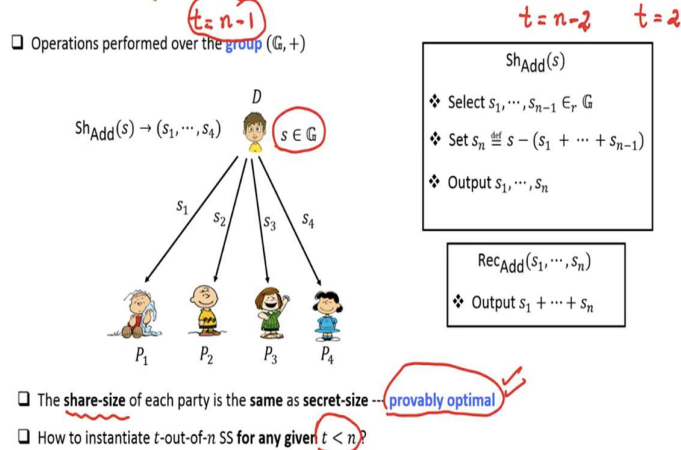
Candidate $s$	Candidate $s_1$
0	3
1	0
2	1
3	2

Now, this time, the view of the parties in  $B$  will be 2 to 1, because this time the shares that they are learning is this. Now, again, you can see that, it could be the case that dealer has shared the secret 0 with equal probability. It could be the case that the dealer has shared the secret 1 with equal probability. It could be the case that dealer has shared the secret 2, with equal probability. It could be the case that dealer has shared the secret 3.

So, just based on these 3 shares, the parties in  $B$  cannot figure out anything. So, in this, all these examples, I am just considering 1 candidate  $B$ . You can take other candidate  $B$ 's, run the sharing algorithm, and then argue by computing these probabilities or just by mapping the candidate  $s$  with the candidate missing piece. And then you can easily find out that indeed it is the case that any subset of  $n - 1$  shareholders, does not learn anything about any underlying secret.

**(Refer Slide Time: 41:11)**

## (n - 1)-out-of-n SS: Discussion



So, this is the summary of your  $n - 1$  out of  $n$  secret-sharing. A very simple additive secret-sharing. Here, what is the share size of each party? Each party as a share, is receiving an element of the group. And secret is also an element of the group. That means, whatever is the size of the underlying secret, the same as the size of each share as well. And it turns out, that is the best we can hope for.

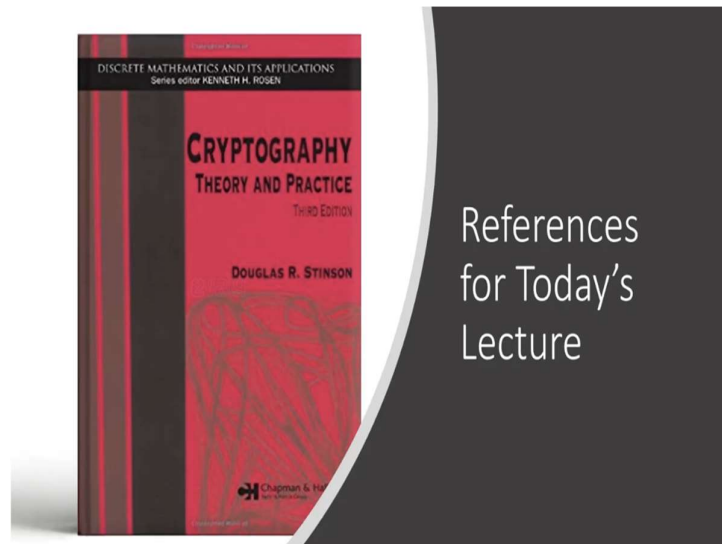
We cannot design a secret-sharing scheme where the size of a share is actually less than the size of the secret. That is the best. That means, the best we can hope for is, that the share size is same as the secret size. And indeed, this secret-sharing scheme ensures that the size of each party, each shareholder is same as the size of the underlying secret. And hence, this is optimal in terms of the share size.

Now, an interesting question is the following. How do we instantiate a threshold  $t$  out of  $n$  secret-sharing for any given  $t < n$ ? What do I mean by that? This additive secret-sharing is a special case of threshold secret-sharing, where your threshold  $t$  was specifically  $n - 1$ . We designed a scheme in such a way that any subset of  $n - 1$  or less number of shareholders, if they come together, they fail to learn anything about the underlying secret.

But what if my  $t$  is equal to say  $n - 2$ ? or what is my say  $t$  is equal to just 2? I want a scheme where any subset of 3 or more shareholders should be able to compute the secret. But any subset of 2 or less number of shareholders, should fail to compute the secret. I cannot run this additive secret-sharing for that case, I have to do something else. So, that means, this additive secret-sharing scheme is tailor made only for the case where  $t$  was exactly  $n - 1$ .

But in general, I might be interested to design a secret-sharing scheme, threshold secret-sharing scheme for any given threshold  $t$  which is strictly less than  $n$ .

**(Refer Slide Time: 43:29)**



So, that will be the focus of our next lecture. Thank you.