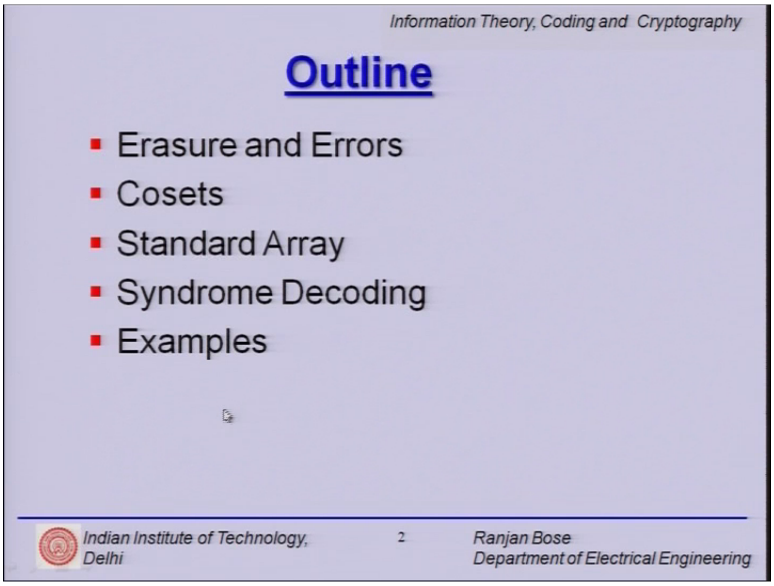
Information Theory, Coding and Cryptography
Dr. Ranjan Bose
Department of Electrical Engineering
Indian Institute of Technology, Delhi

Module – 17
Linear Block Codes
Lecture – 17

(Refer Slide Time: 00:32)



Information Theory, Coding and Cryptography

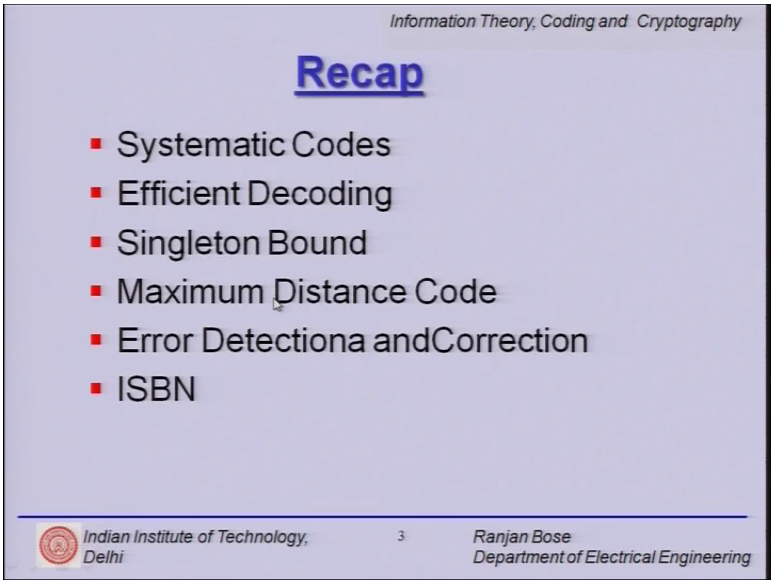
Outline

- Erasure and Errors
- Cosets
- Standard Array
- Syndrome Decoding
- Examples

Indian Institute of Technology, Delhi
2
Ranjan Bose
Department of Electrical Engineering

Hello, and welcome to our next module on linear block codes. Let us begin with a brief outline. We will start with erasures and errors as encountered by the bit stream sent over an unreliable channel. Then we will exploit the notion of cosets, and standard array, and these are important from classical perspectives. Then we will go on to looking at what is syndrome decoding, what do we understand by syndromes and then, we will look at some examples so that is the agenda for today’s lecture.

(Refer Slide Time: 01:04)



Information Theory, Coding and Cryptography

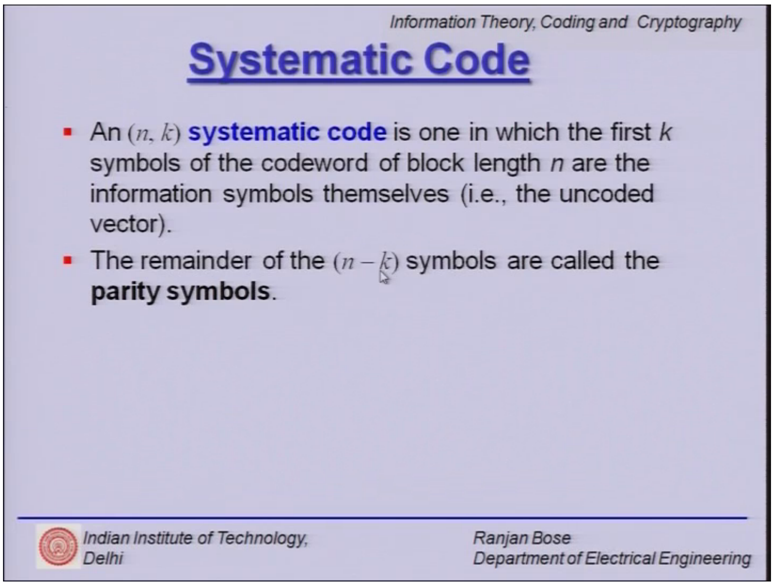
Recap

- Systematic Codes
- Efficient Decoding
- Singleton Bound
- Maximum Distance Code
- Error Detectiona andCorrection
- ISBN

Indian Institute of Technology, Delhi 3 Ranjan Bose Department of Electrical Engineering

But we start with a quick recap, we have already done systematic codes. We have talked about efficient decoding, singleton bound. We have talked about maximum distance codes and error connection and detection.

(Refer Slide Time: 01:20)



Information Theory, Coding and Cryptography

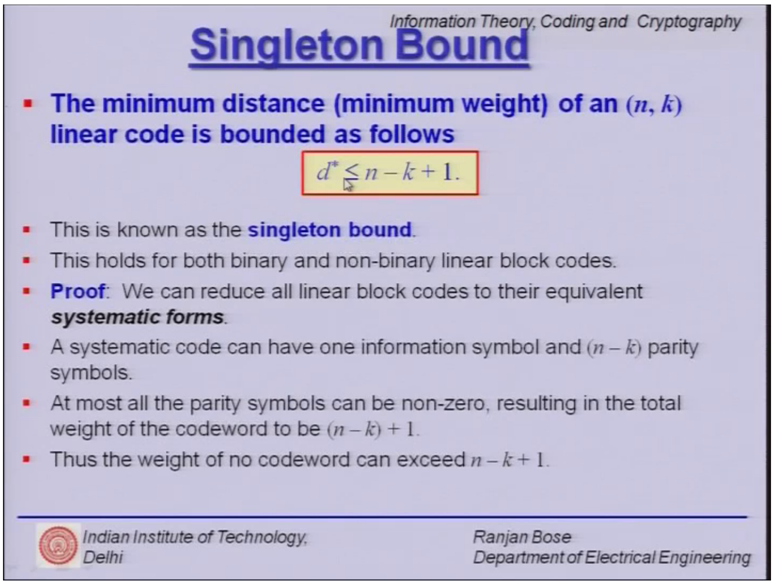
Systematic Code

- An (n, k) **systematic code** is one in which the first *k* symbols of the codeword of block length *n* are the information symbols themselves (i.e., the uncoded vector).
- The remainder of the $(n-k)$ symbols are called the **parity symbols**.

Indian Institute of Technology, Delhi Ranjan Bose Department of Electrical Engineering

So, let us look at quickly, what we meant by saying a systematic code. And n comma k systematic code is one in which the first k symbols of the codeword of the block length n are information symbols themselves, that is the uncoded vector. The remainder n minus k are the parity symbols.

(Refer Slide Time: 01:44)



Information Theory, Coding and Cryptography

Singleton Bound

- **The minimum distance (minimum weight) of an (n, k) linear code is bounded as follows**

$$\boxed{d^* \leq n - k + 1.}$$

- This is known as the **singleton bound**.
- This holds for both binary and non-binary linear block codes.
- **Proof**: We can reduce all linear block codes to their equivalent ***systematic forms***.
- A systematic code can have one information symbol and $(n - k)$ parity symbols.
- At most all the parity symbols can be non-zero, resulting in the total weight of the codeword to be $(n - k) + 1$.
- Thus the weight of no codeword can exceed $n - k + 1$.

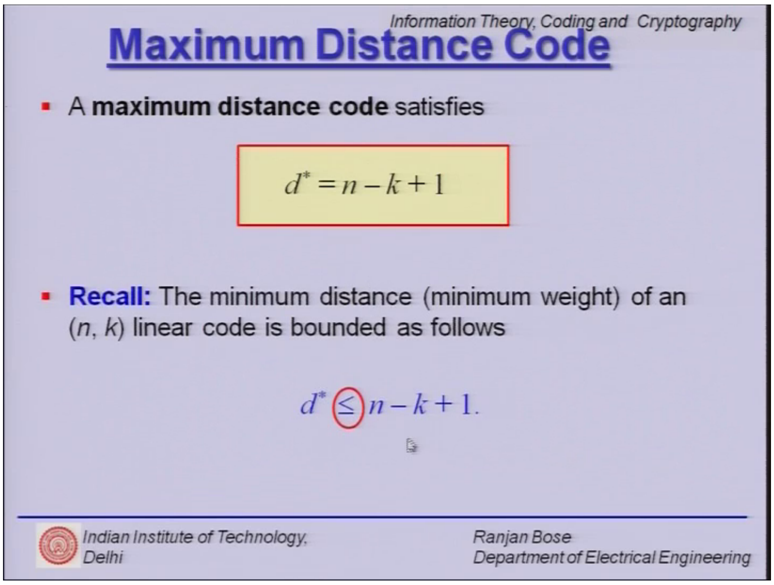
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

We looked at the singleton bound, where the minimum distance also the minimum weight for a linear block code is given by d star, which is upper bounded by n minus k plus 1. This is the singleton bound. This holds true both for binary and non-binary linear block codes. And we did a quick proof of that by reducing the generator matrix into its systematic form. And you have the identity matrix, and the parity matrix at the end.

And we saw that each of the rows of this systematic form generator matrix is a valid code word, and the weight of any code word can at best be n minus k plus 1. So, no codeword can have a weight that can exceed n minus k plus 1. And therefore the minimum distance, which is also equal to the minimum weight can never be greater than n minus k plus 1. Hence, d star is less than or equal to n minus k plus 1, this is called the singleton bound.

(Refer Slide Time: 02:57)



Information Theory, Coding and Cryptography

Maximum Distance Code

- A **maximum distance code** satisfies

$$\boxed{d^* = n - k + 1}$$

- **Recall:** The minimum distance (minimum weight) of an (*n*, *k*) linear code is bounded as follows

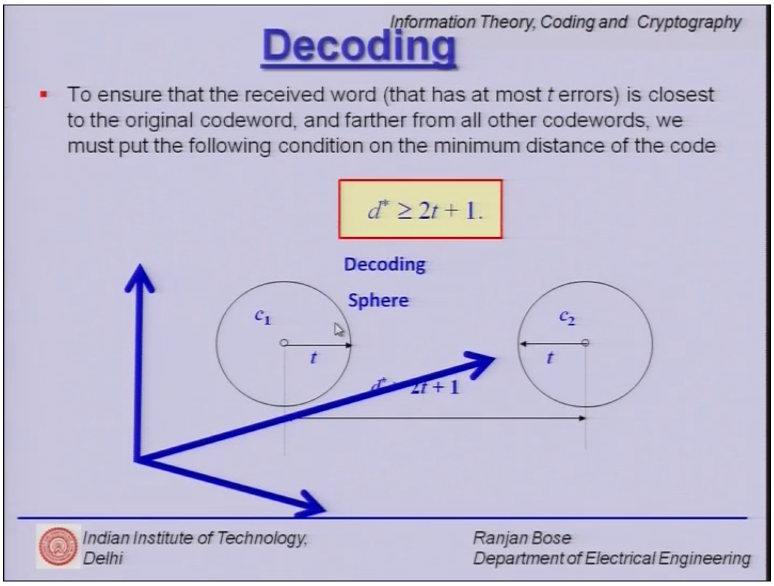
$$d^* \leq n - k + 1.$$

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

Based on that we defined a maximum distance code and the maximum distance code satisfies the equality d star is indeed equal to n minus k plus 1. So, in some sense, it is a good property to have a maximum distance code is a good code, because you maximize the distance minimum distance given the constraints of n and k fine. And we got this equality by putting the upper bound at equal to 1.

(Refer Slide Time: 03:38)



Information Theory, Coding and Cryptography

Decoding

- To ensure that the received word (that has at most *t* errors) is closest to the original codeword, and farther from all other codewords, we must put the following condition on the minimum distance of the code

$$\boxed{d^* \geq 2t + 1.}$$

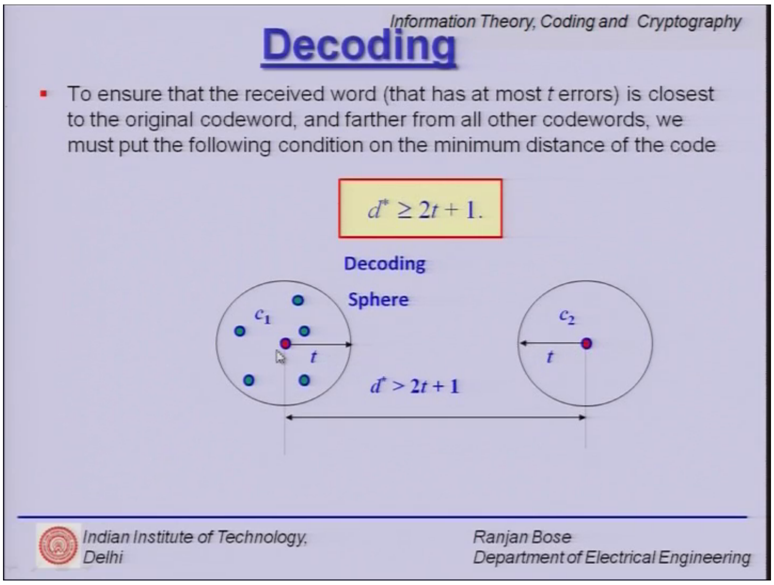
Decoding Sphere

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

Now, we look at decoding part. So, we have moved from the transmission side to the receiving side, and we would like to see, t errors getting corrected in a n symbol long codeword. So, we established that in order to correct t errors, you have the condition d star the minimum distance should be greater than or equal to 2 times t plus 1. And the rationale behind it can be seen geometrically. So, suppose you have an n dimensional space, and every point in this space is an n tuple. So, every code word is a point in space.

(Refer Slide Time: 04:38)



Information Theory, Coding and Cryptography

Decoding

- To ensure that the received word (that has at most *t* errors) is closest to the original codeword, and farther from all other codewords, we must put the following condition on the minimum distance of the code

$$\boxed{d^* \geq 2t + 1.}$$

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

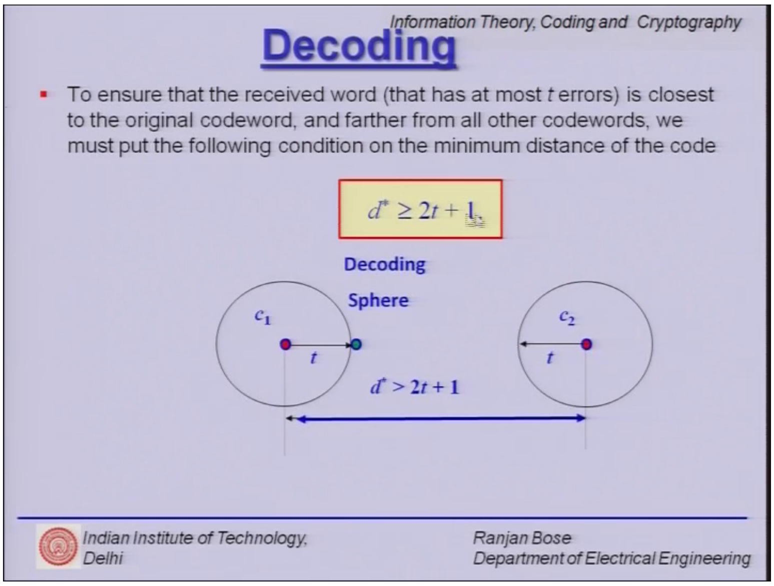
So, let us put c 1 and c 2. And around this c 1 and c 2, we draw 2 spheres, and the radius of this is t that means, all the code words are words sitting inside this are less than or equal to t hamming distance apart. So, each and every point within this sphere is a hamming distance t or less. And the points sitting on the circumference is actually at a hamming distance t. This is the decoding sphere of radius t.

Similarly, I can draw another decoding sphere of radius t for c 2. And the premise is that we would decode any word received within this decoding sphere, and we will make it map it back to c 1 here. Any received vector a world, which is found within this sphere. What do you mean by finding word within this sphere, any point within this sphere represents an end symbol long codeword, and we can receive any possible codeword or word at the receiver, which is n symbol long. We find out the hamming distance, and if it is within t, then we map it back to c 2, and this is the nearest neighbor decoding.

Now, what we want is to ensure that every time if t errors or less occur, then we should be uniquely be able to decode it that is that none of these fears should be overlapping. What do we mean by overlapping, if the overlap, then there are some points, which are common to both the spheres, and we will never be able to uniquely decode it back either to c 1 or to c 2 or to any other codeword.

So, what we want is non-overlapping, to ensure them they are non-overlapping. These centers of these spheres must be at least 2t apart only, then they cannot overlap, but that also means, if there is exactly 2t they are touching. So, some points on the circumference would still be in ambiguity. So, we put that additional measure of 1, because distance hamming distance is always an integer, so d star should be greater than or equal to 2 t plus 1 in order to ensure that t could errors in the codewords are always compensated.

(Refer Slide Time: 07:11)



Information Theory, Coding and Cryptography

Decoding

- To ensure that the received word (that has at most t errors) is closest to the original codeword, and farther from all other codewords, we must put the following condition on the minimum distance of the code

$$\boxed{d^* \geq 2t + 1}$$

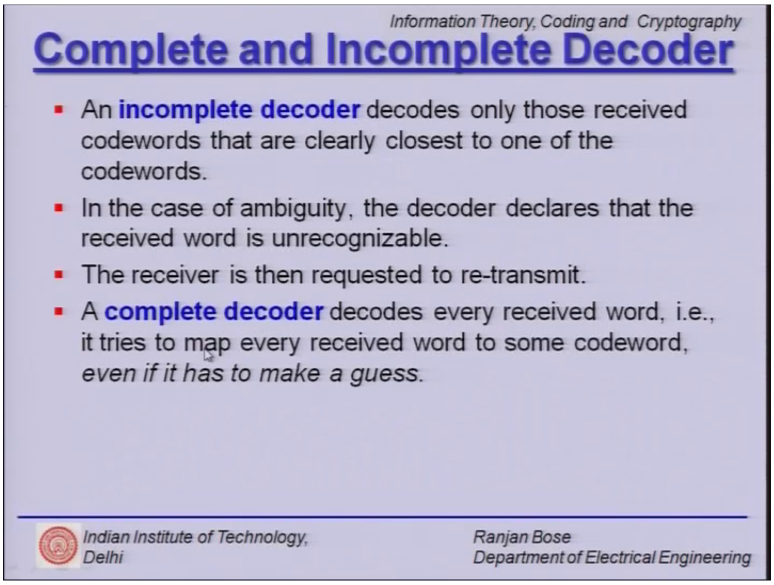
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

For so we can always correct for t errors provided d star is greater than or equal to 2 t plus 1. In this diagram, we have put greater than necessarily, because they have separated these spheres even further. So, this is the condition, it is telling us a very important thing.

It tells us that d star indeed is a single most important parameter for linear block codes, because it is directly linked to the error correcting capability, and that is what we started out with, we would like to correct a certain number of errors. So, if you can give me a d star, which is large enough I can correspondingly correct the errors depending upon the number of t errors that satisfies an inequality.

(Refer Slide Time: 08:05)



Information Theory, Coding and Cryptography

Complete and Incomplete Decoder

- An **incomplete decoder** decodes only those received codewords that are clearly closest to one of the codewords.
- In the case of ambiguity, the decoder declares that the received word is unrecognizable.
- The receiver is then requested to re-transmit.
- A **complete decoder** decodes every received word, i.e., it tries to map every received word to some codeword, *even if it has to make a guess.*

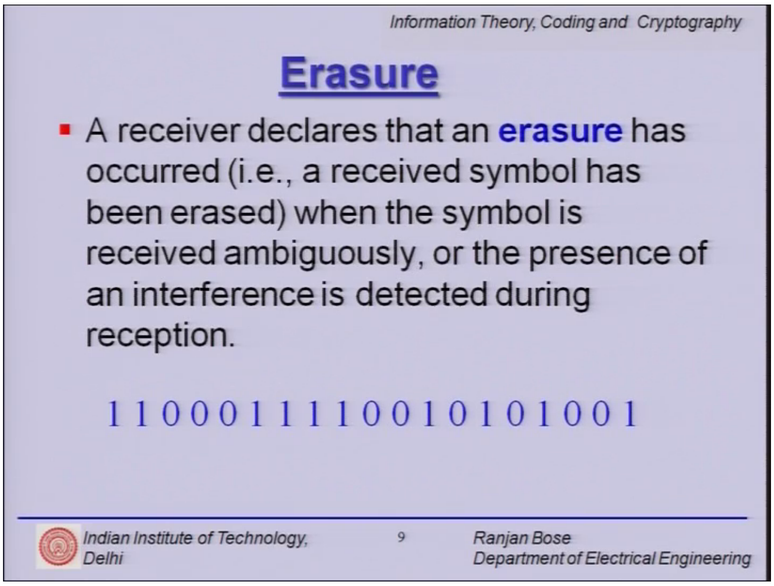
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

Now, an incomplete decoder decodes only those received codewords or words that are clearly closest to the one of the codewords, there is no in ambiguity. However, in the case there is ambiguity, the decoder declares that the received word is unrecognizable, and requests for a re-transmission. Such a strategy is under incomplete decoding strategy.

However, a complete decoder decodes every received word the word, regardless of whether that is ambiguous or not, so it may have to make a guess once in a while. And such decoder is called an a complete decoder.

(Refer Slide Time: 08:51)



Information Theory, Coding and Cryptography

Erasure

- A receiver declares that an **erasure** has occurred (i.e., a received symbol has been erased) when the symbol is received ambiguously, or the presence of an interference is detected during reception.

1 1 0 0 0 1 1 1 1 0 0 1 0 1 0 1 0 0 1

Indian Institute of Technology, Delhi

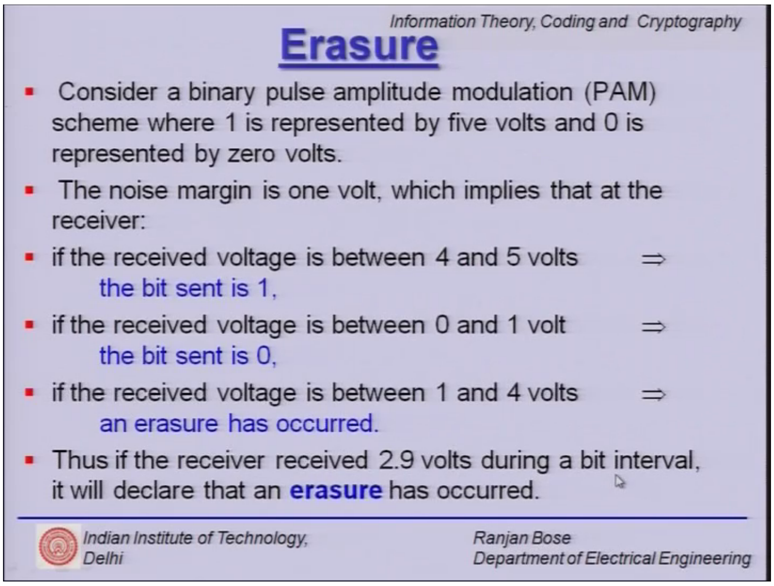
9

Ranjan Bose
Department of Electrical Engineering

Now, we start with another kind of problem that we encounter, while transmitting a series of bit streams, or symbol streams over an reliable channel. So, if the channel is unreliable, it may not only lead to errors, it can also give you erasures. What do we mean by erasure, I receive a declares that an erasure has occurred when a symbol is received ambiguously or their presence of an interference is detected during the reception.

So, instead of guessing is it a 1 or a 0 what you do is, say look some of these bits kind of did not come through correctly, and for me they have been erased. So, we do not really know, where that a 1 was received or a 0 was received. Now, we are we not saying it is an error, we simply say they erased. So, there is a difference between erasure and in being in error.

(Refer Slide Time: 09:53)



Information Theory, Coding and Cryptography

Erasure

- Consider a binary pulse amplitude modulation (PAM) scheme where 1 is represented by five volts and 0 is represented by zero volts.
- The noise margin is one volt, which implies that at the receiver:
- if the received voltage is between 4 and 5 volts ⇒ the bit sent is 1,
- if the received voltage is between 0 and 1 volt ⇒ the bit sent is 0,
- if the received voltage is between 1 and 4 volts ⇒ an erasure has occurred.
- Thus if the receiver received 2.9 volts during a bit interval, it will declare that an **erasure** has occurred.

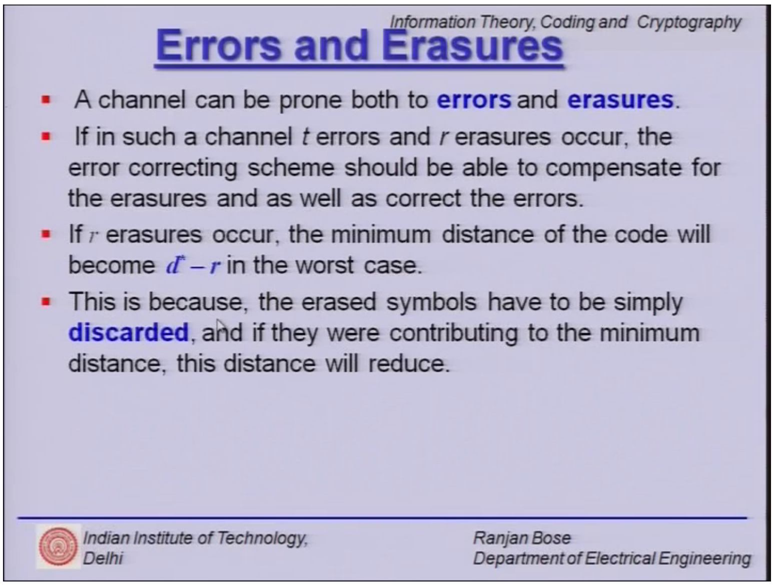
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

So, I will give you a practical example, where erasure can actually occur. Considered a binary pulse amplitude modulation, a PAM scheme, where 1 is represented by five volts and 0 is represented by zero volts.

Now, of course, we always put in a noise margin, so suppose one volt is the noise margin. So, what do we do, we have a simple decoding strategy at the receiver, what we say is that if I receive a voltage between 4 and 5 volts, we say hey 1 was sent. At the other hand if we get between 0 and 1, we declare bit 0 was being sent. But, what do we do in between well if it is between 1 and 4, we say look an erasure has occurred, I do not want to take a chance I would like to declare it not an error, but in erasure. So, if 2.9 volt is received during a bit interval we say look that bit has been erased.

(Refer Slide Time: 10:55)



Information Theory, Coding and Cryptography

Errors and Erasures

- A channel can be prone both to **errors** and **erasures**.
- If in such a channel *t* errors and *r* erasures occur, the error correcting scheme should be able to compensate for the erasures and as well as correct the errors.
- If r erasures occur, the minimum distance of the code will become $d^* - r$ in the worst case.
- This is because, the erased symbols have to be simply **discarded**, and if they were contributing to the minimum distance, this distance will reduce.

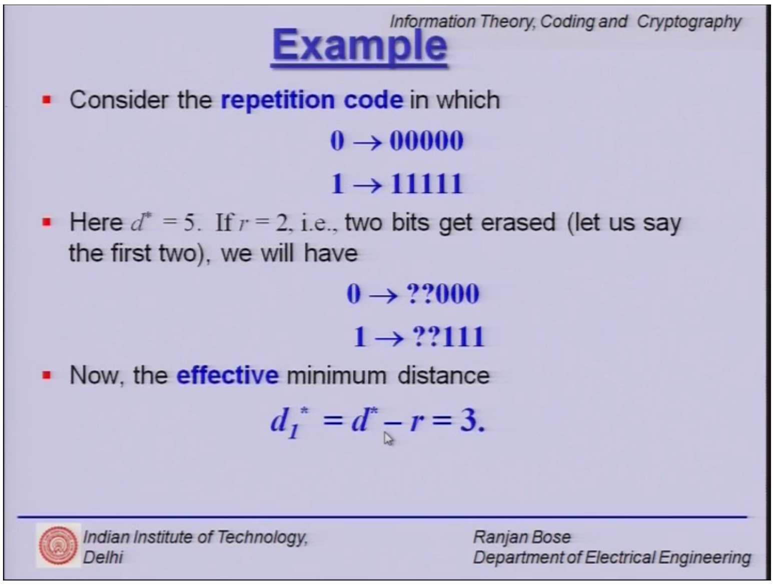
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

So, how do we handle this, it is clearly different from errors. A channel is prone both to errors and erasures. So, if in a channel, we are compensating or planning to correct t errors and r erasures, then the corresponding error correcting scheme should be able to handle both t errors and r erasures. So, if r erasures occur, it simply means that have r fewer symbols to work with, which means that the effective minimum distance actually becomes d star minus r that is the worst case, and we are looking at the worst case scenario.

So, if your channel is producing r erasures in a n symbol long codeword, then all we have left with us is d star minus r in the worst case, this is because a symbol have to simply be discarded, and they were contributing to the minimum distance, then that minimum distance will reduce. What is the minimum distance, it is the minimum weight. So, if the erasures happen to happen in those non-zero elements, then your d star actually reduces.

(Refer Slide Time: 12:20)



Information Theory, Coding and Cryptography

Example

- Consider the **repetition code** in which

$$0 \rightarrow 00000$$
$$1 \rightarrow 11111$$

- Here $d^* = 5$. If $r = 2$, i.e., two bits get erased (let us say the first two), we will have

$$0 \rightarrow ??000$$
$$1 \rightarrow ??111$$

- Now, the **effective** minimum distance

$$d_1^* = d^* - r = 3.$$

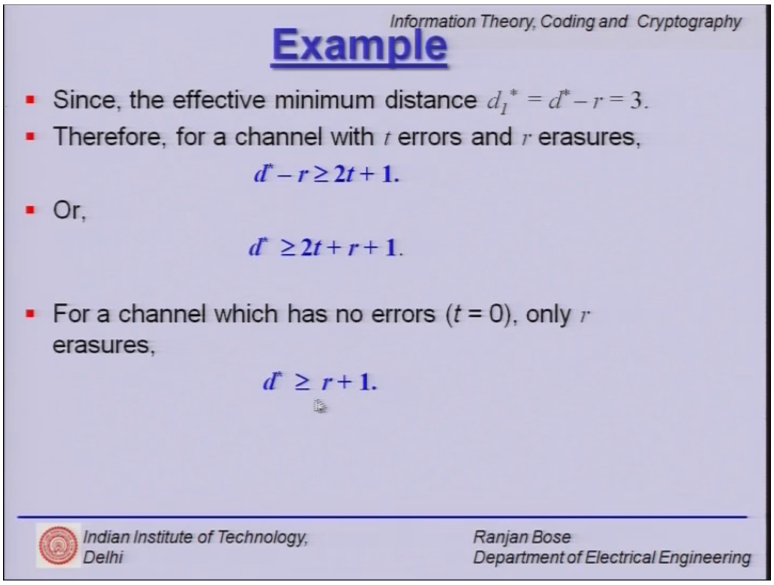
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

Consider this repetition code. So, 0 is repeated five times, and so is 1. So, whenever I have to send 0, I send 0 0 0 0 0; and whenever I have to send a 1, I send 1 1 1 1 1. So, clearly d star is 5, the minimum distance or minimum weight is 5. Suppose, we have r is equal to 2 that is the erasure two of the bits get erased. And for the sake of discussion only, we say that the first two bits are erased, we just happen to miss the first two bits of the transmission for whatever reason, and then the next three bits are cut. Same with 1 I happen to miss the first two elements of this codeword, but rest we catch.

So, the effective minimum distance has now, you can obviously see is 3, and we have to work with this. Based on this, we have to decide, where there was 0 was sent, or a 1 was sent. So, erasure can really be detrimental to error correction, because effectively a d star has gone down.

(Refer Slide Time: 13:43)



Information Theory, Coding and Cryptography

Example

- Since, the effective minimum distance $d_1^* = d^* - r = 3$.
- Therefore, for a channel with t errors and r erasures,

$$d^* - r \geq 2t + 1.$$

- Or,

$$d^* \geq 2t + r + 1.$$

- For a channel which has no errors ($t = 0$), only r erasures,

$$d^* \geq r + 1.$$

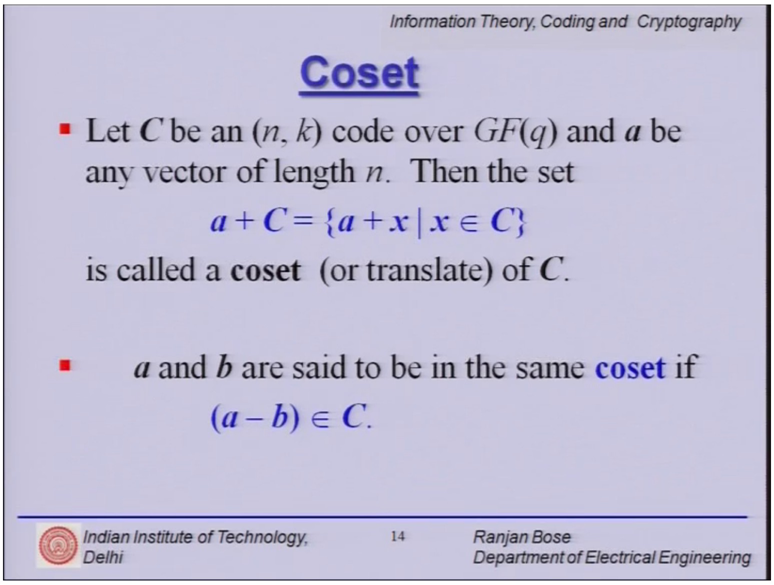
Indian Institute of Technology,
Delhi

Ranjan Bose
Department of Electrical Engineering

So, we will work with this reduced minimum distance, so d star equal to 3, now will allow me to correct for only one error. Earlier d star was 5 before we had an erasure, so we it was able to correct two errors, so this erasures have resulted in decreased capability of error correction.

So, what is why, new constraint d star minus r, if the channel is producing r erasures should be greater than or equal to 2 t plus 1, in order to correct for t errors. This is what we write for a channel, which has both t errors and r erasures; or you can put r on the right hand, side and put d star greater than or equal to 2 t plus r plus 1, so r clearly, it is up into the number of errors you can correct. Now, we can also have a channel, which does not have errors, it just has erasures, so t is equal to 0, but only r erasures. And in that case, your minimum distance should be greater than r plus 1.

(Refer Slide Time: 15:06)



Information Theory, Coding and Cryptography

Coset

- Let C be an (n, k) code over $GF(q)$ and $\boldsymbol{a}$ be any vector of length n. Then the set

$$a + C = \{a + x \mid x \in C\}$$

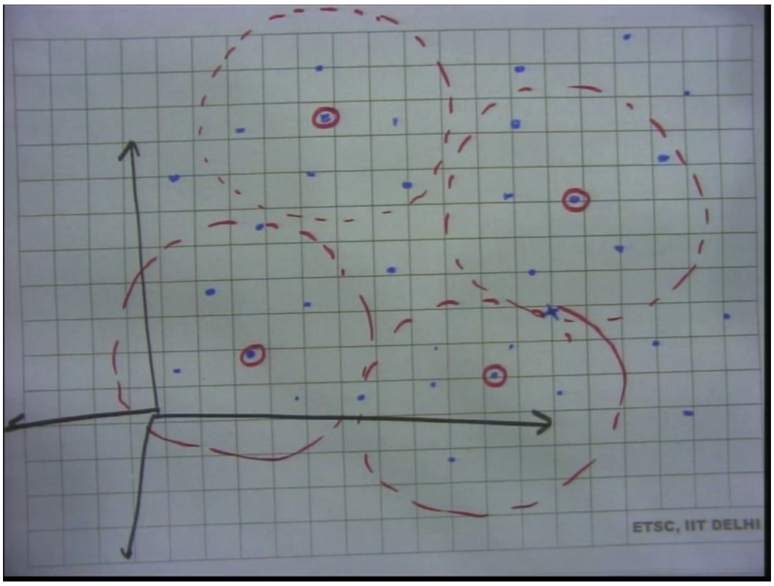
is called a **coset** (or translate) of C.

- $\boldsymbol{a}$ and $\boldsymbol{b}$ are said to be in the same **coset** if $(a - b) \in C$.

Indian Institute of Technology, Delhi 14 Ranjan Bose Department of Electrical Engineering

Now, let us move on and look at things a little differently from the decoding perspective. The next few slides will look at what are the ways to decode a received vector, which possibly is an error. So, please remember, what we have done is chosen a small subset of vectors within the space, and we have declared them as valid codewords.

(Refer Slide Time: 15:51)



Now, if you look at it diagrammatically, we have an n dimensional space. And this n dimensional space has several points. In fact, if it is binary, then we will have to respect by n points. Now, of these we have to choose to respect k points to correspond to your valid codewords. So, clearly many of the points are left out. In fact, most of the points are left out.

The question before us is that how can we arrange these different vectors, because please remember each point is a vector. How do we arrange them in a logical manner, so that decoding becomes easy. What has been the problem of decoding, well decoding logically says that each of the codeword has a sphere around it, which is the decoding sphere. And hopefully they are not overlapping, at least overlapping in the sense that no common vectors are present in both overlaps, so no such vectors are present.

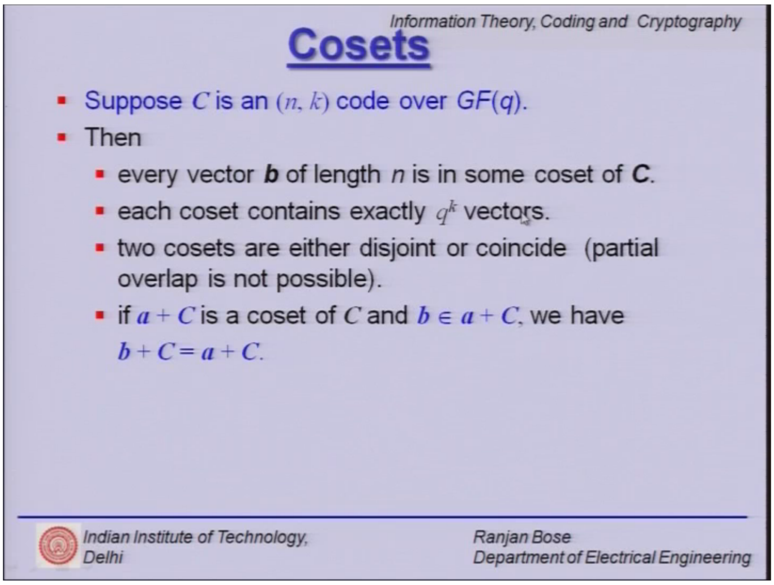
Now, in this space we would like to organize all of these points or vectors in a table format, which will quickly help us decode. And clearly, it has to be done with respect to the hamming distance. So, would not it be great, if all the vectors within this sphere are listed in one column.

Similarly, I have another look-up table, which says that all the vectors or points closes to this codeword are listed together and so and so forth. This organization will quickly help me do the nearest neighbor decoding. And to do this logically, we would now introduce this concept of cosets. So, we are talking about sets and cosets all right. Please note, that there are many points outside some of this, they are not covered by any of these spheres all right.

So, we go back to our slide, and let us start with a code C, n comma k code, which is defined over G F q. So, we are talking not only for binary case, but for q symbols in general. Now, let a be any vector of length n. So, clearly we are talking about from that code space there are a subset of codewords, and a could be any other vector of length n. Since, it is of length n, it is also another point.

Then we can form the set a plus c is defined as a plus x, where x is an element of C, so x is a valid code word. So, I take this a and I add it up, and if it is over G F q, we use the addition tables of G F q. This set is called the coset or translate of C ok, it is clearly a translation. So, it is a translate of C. Now, a and b are said to be in the same coset. What are a and b, both are vectors of length n. So, a and b are said to be the same coset, if a minus b is also an element of C.

(Refer Slide Time: 20:16)



Information Theory, Coding and Cryptography

Cosets

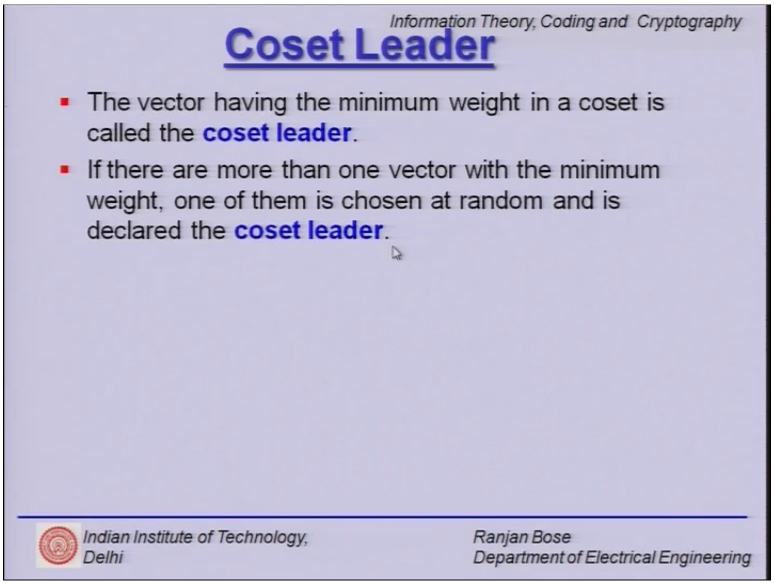
- Suppose C is an (n, k) code over $GF(q)$.
- Then
  - every vector $\boldsymbol{b}$ of length n is in some coset of $\boldsymbol{C}$.
  - each coset contains exactly q^k vectors.
  - two cosets are either disjoint or coincide (partial overlap is not possible).
  - if $\boldsymbol{a} + C$ is a coset of C and $\boldsymbol{b} \in \boldsymbol{a} + C$, we have $\boldsymbol{b} + C = \boldsymbol{a} + C$.

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

So, again we start with this n minus k code over G F q. Then every vector b of length n is in some coset of C all right. Each coset contains exactly q raise power k vectors. And two cosets are either disjoint or coincide, because partial overlap is not possible. If a plus C is a coset of C, and b is an element of a plus C we have b plus C is equal to a plus C. So, what it tells us is that we are forming groups, we are forming sets right and logically partitioning the space into cosets.

(Refer Slide Time: 21:10)



Information Theory, Coding and Cryptography

Coset Leader

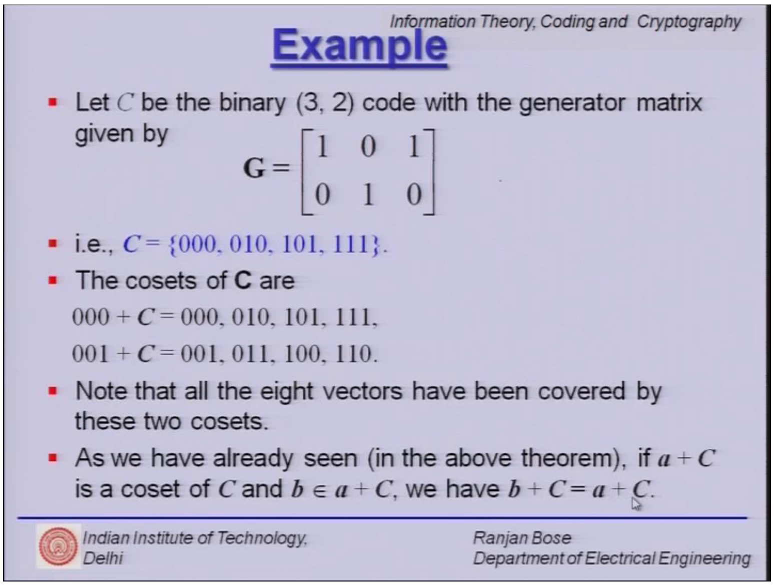
- The vector having the minimum weight in a coset is called the **coset leader**.
- If there are more than one vector with the minimum weight, one of them is chosen at random and is declared the **coset leader**.

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

Now, what to be mean by a coset leader. The vector having the minimum weight in a coset is called the coset leader. We have already figured out how to make a coset, which is a translate. If they are more than one vector with minimum weight, then one of them is chosen at random, and is declared a coset leader. We will use this notion of coset leader to do our decoding shortly.

(Refer Slide Time: 21:39)



Information Theory, Coding and Cryptography

Example

- Let C be the binary (3, 2) code with the generator matrix given by

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- i.e., $C = \{000, 010, 101, 111\}$.
- The cosets of **C** are

$000 + C = 000, 010, 101, 111,$

$001 + C = 001, 011, 100, 110.$

- Note that all the eight vectors have been covered by these two cosets.
- As we have already seen (in the above theorem), if $a + C$ is a coset of C and $b \in a + C$, we have $b + C = a + C$.

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

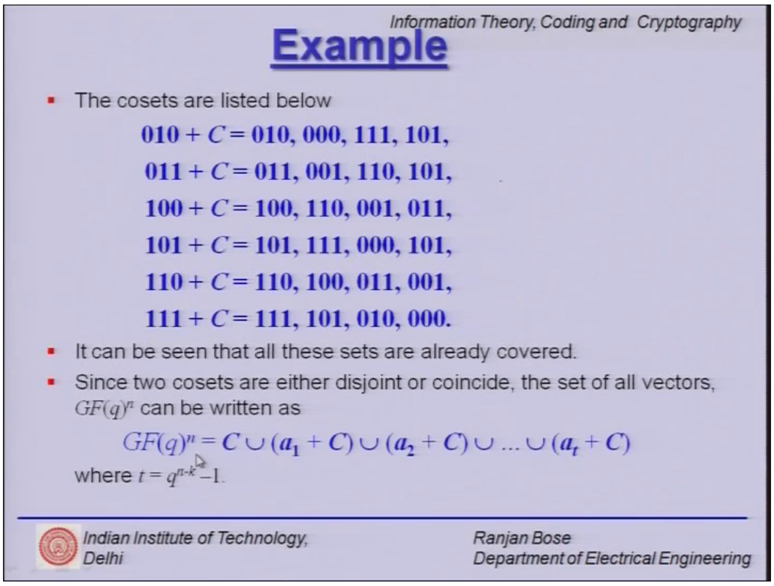
So, let us look at a quick example. Let us look at a 3 comma 2 code, so it is a linear code, and therefore, we have a generator matrix representation of it, so k is 2 and n is 3, and this is a simple linear block code. So, there are four code words 0 0 0, 0 1 0, 1 0 1, and 1 1 1, they are formed by this. You can do a quick sanity check, 0 0 0 necessarily must be there.

Then that two rows of G must be valid codewords, so 1 0 1, and 0 1 0 are present. And sum of any two codewords is a valid codeword. So, if you add these two, I get 1 1 1, so that is also a valid codeword, clearly I use this basis vector to generate 1 1 1. And then, I can satisfy all other properties and therefore, this linear block code is obtained.

Now, we come to the definition of cosets, and we have defined the cosets for this. So, let us look at these translates. So, I take this vector a n is 3, so this is a vector of length 3 plus c. What is C, well C itself is a set. So, the coset, when you add it up it is the same set. So, this is one coset, but that is not the only one, because if my random vector a was 0 0 1, and I add them up, so that translates are obtained as follows.

And we have the all the possible eight vectors, because n is equal to 3, so 2 is (Refer Time: 23:32) 3s eight possible vectors can be formed. Just these two cosets have covered the entire space. So, what have we done, we have partitioned the entire space into these two subsets the cosets. So, all the eight vectors are covered by these cosets. So, if a plus C is a coset of C and b is an element of a plus C, then we have b plus C is equal to a plus C.

(Refer Slide Time: 24:03)



Information Theory, Coding and Cryptography

Example

- The cosets are listed below

$$010 + C = 010, 000, 111, 101,$$
$$011 + C = 011, 001, 110, 101,$$
$$100 + C = 100, 110, 001, 011,$$
$$101 + C = 101, 111, 000, 101,$$
$$110 + C = 110, 100, 011, 001,$$
$$111 + C = 111, 101, 010, 000.$$

- It can be seen that all these sets are already covered.
- Since two cosets are either disjoint or coincide, the set of all vectors, $GF(q)^n$ can be written as

$$GF(q)^n = C \cup (a_1 + C) \cup (a_2 + C) \cup \ldots \cup (a_t + C)$$

where $t = q^{n-k} - 1$.

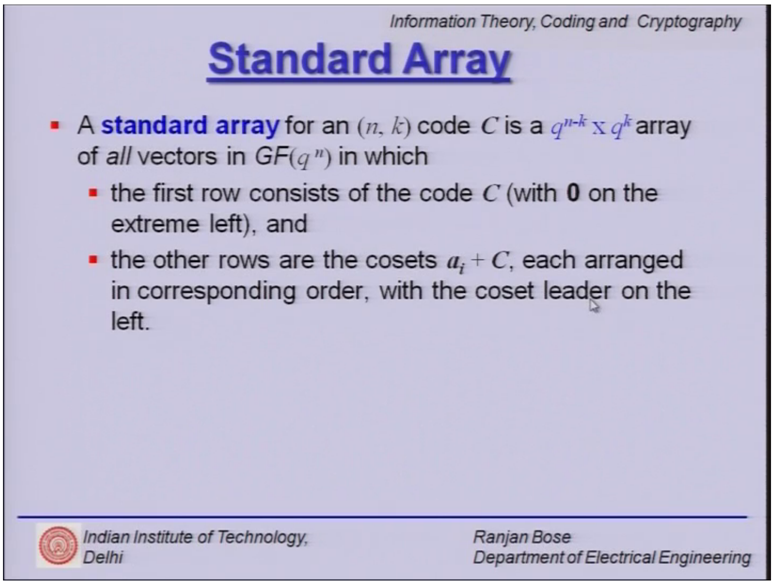
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

So, if I list down all the possible vectors now beyond that. So, if I have 0 1 0 as the a vector and add it up I get this subset or coset; if I have the translates of 0 1 1 I get this and so on. So, it is a very mechanical process and I can generate this things. And for different a vectors, I get repetition, so 1 1 1 was here in this, and it was here also right.

So, these sets are already covered. Since, two cosets are either disjoint or coincide, the set of all vectors, over G F q can be written as these. So, in the last slide we saw that a 1 was 0 0 0, and a 2 was 1 1 1, and a 0 1 0 0 1, and they covered the entire space of G F q n.

(Refer Slide Time: 25:07)



Information Theory, Coding and Cryptography

Standard Array

- A **standard array** for an (n, k) code C is a $q^{n-k} \times q^{k}$ array of *all* vectors in $GF(q^{n})$ in which
  - the first row consists of the code C (with **0** on the extreme left), and
  - the other rows are the cosets $a_i + C$, each arranged in corresponding order, with the coset leader on the left.

Indian Institute of Technology,
Delhi

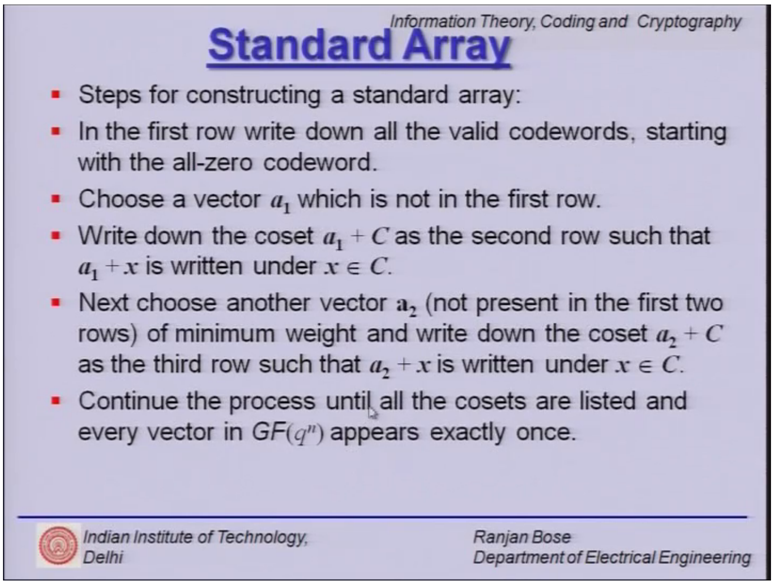
Ranjan Bose
Department of Electrical Engineering

Now, we will use these cosets to define something called a standard array. And remember, our final aim is to find an efficient decoding strategy, so we are moving towards that slowly. So, what is a standard array, it is a table clearly. For an n comma code C is a q raise power n minus k cross q raise power k array of all vectors in G F q n. So, we are smartly going to partition or tabulate all the vectors present in G F q n.

Why are we dealing with all vectors of G F q n, because whenever we received a vector at the receiver, it is of size n. So, we are going to receive one of the vectors of over G F q n. And our aim is given that vector, and it could be an error. We have to find out, which is the most likely code word that was sent. And this standard array is supposed to help us do that job.

So, how do we make a standard array? The first row consists of the code C all right. The other rows are the cosets a i plus C, each arranged in a corresponding order, with the coset leader on the left; remember coset leader has the minimum weight all right. The set of vectors all have vectors of different lengths, so different weights, so you have coset leader with the minimum weight.

(Refer Slide Time: 26:45)



Information Theory, Coding and Cryptography

Standard Array

- Steps for constructing a standard array:
- In the first row write down all the valid codewords, starting with the all-zero codeword.
- Choose a vector a_1 which is not in the first row.
- Write down the coset $a_1 + C$ as the second row such that $a_1 + x$ is written under $x \in C$.
- Next choose another vector $\mathbf{a}_2$ (not present in the first two rows) of minimum weight and write down the coset $a_2 + C$ as the third row such that $a_2 + x$ is written under $x \in C$.
- Continue the process until all the cosets are listed and every vector in $GF(q^n)$ appears exactly once.

Indian Institute of Technology, Delhi

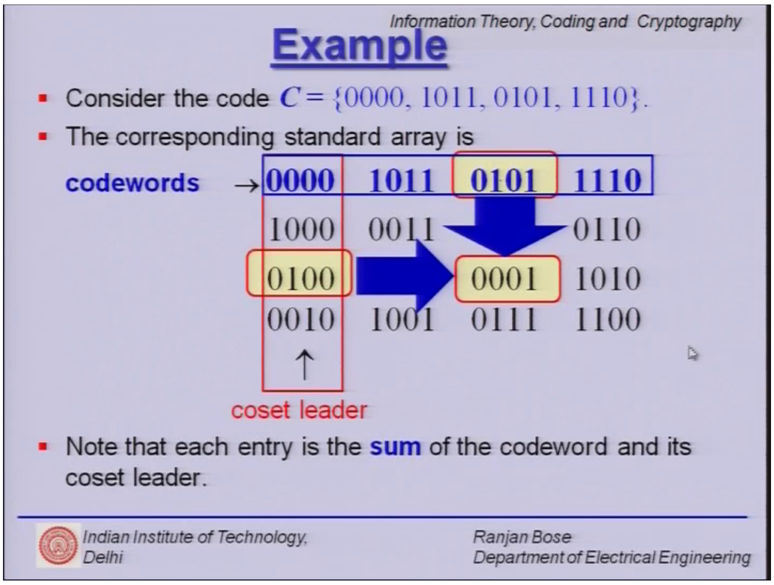
Ranjan Bose
Department of Electrical Engineering

So, how do we construct a standard array. In the first row write down all the valid codewords; starting with the all-zero codeword. Choose a vector a i, which is not in the first row. Then write down the cosets a 1 plus C as a second row, such that a 1 plus x is written under x, which is an element of C.

Next choose another vector a 2, because I can keep on forming my cosets. But, this a 2 is should be not present in the first 2 rows, because we are gradually going to span the entire space. But, the first vector should be of the minimum weight, because it has to be the coset reader. And write down the coset for a 2 plus C as a third row, such that a 2 plus C is written under x element of C.

Continue the process until all the cosets are listed, and each and every vector in G F q n appears exactly one. So, we would have logically partitioned entire space, and written down in a tabular manner, each and every vector of G F q n. And each vector will appear only once. And each of the rows will be led by the coset leader.

(Refer Slide Time: 28:05)



Information Theory, Coding and Cryptography

Example

- Consider the code $C = \{0000, 1011, 0101, 1110\}$.
- The corresponding standard array is

codewords →

0000	1011	0101	1110
1000	0011		0110
0100		0001	1010
0010	1001	0111	1100

↑ coset leader

- Note that each entry is the **sum** of the codeword and its coset leader.

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

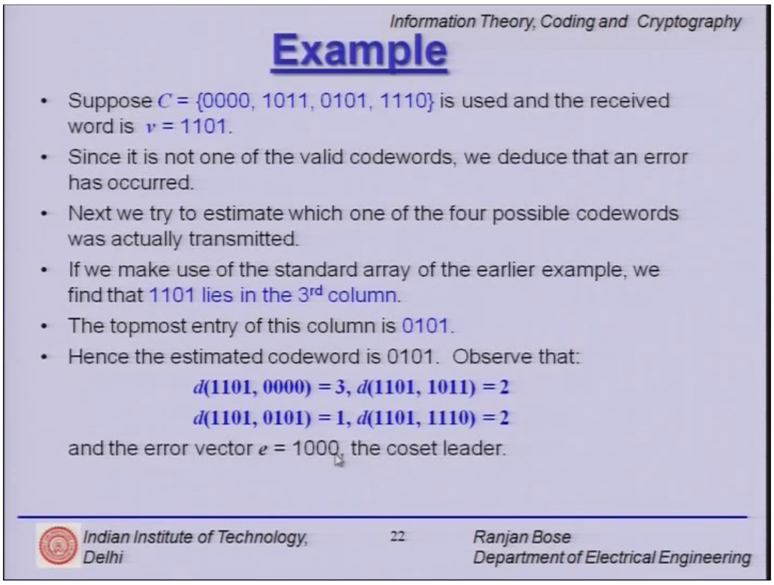
Let us understand this using a simple example. So, we have a same code, code is a set of codewords, we have this 3 comma 2 code. So, or in this case, it is n is four, so it is 4 comma 2 code, and you have these four codewords available.

So, if you want to make a standard array, you first write the codewords here, so 0 0 0 0 and by definition should be the first codeword to be listed out. And then, we have the other three codewords listed out, that forms my first row that is pretty simple. And then, we form the subsequent row by, so if my translate a i is 1 0 0 0 I add it up, and I add it up, and I add it up, and I add it up, and I get this second row ok. And they are completely different from the first one.

And similarly, if I have my a 2 as 0 1 0 0 and I add them up, so I get 0 1 0 0 here. And then, if I add 0 1 0 0 to this one, I get 1 1 1 1 and so and so forth. Similarly, if my a 3 is 0 0 1 0, I can add 2 corresponding code words, and I get the translates or the cosets. So, I have got these four cosets. But, first column is the coset leader, because you can see that the weight is minimum ok. The weight of each of them is unity, rest all could be possibly greater than 1.

So, we have a standard array with the coset leader, and these are the cosets. It should be very clear that this minimum weight leads the pack. And of course, how do we started it off, we started with the codewords. So, the starting point is the codeword, and we have formed this standard array. Note that each entry is the sum of the codeword, and it is coset leader. We have done that, in fact, that is how we have constructed it. So, suppose you wanted to look at this element, it should necessarily be the sum of the codeword and the coset leader, and so and so forth for each and every entry inside this array standard array.

(Refer Slide Time: 30:45)



Information Theory, Coding and Cryptography

Example

- Suppose C = {0000, 1011, 0101, 1110} is used and the received word is v = 1101.
- Since it is not one of the valid codewords, we deduce that an error has occurred.
- Next we try to estimate which one of the four possible codewords was actually transmitted.
- If we make use of the standard array of the earlier example, we find that 1101 lies in the 3rd column.
- The topmost entry of this column is 0101.
- Hence the estimated codeword is 0101. Observe that:

$$d(1101, 0000) = 3, d(1101, 1011) = 2$$
$$d(1101, 0101) = 1, d(1101, 1110) = 2$$

and the error vector e = 1000, the coset leader.

Indian Institute of Technology, Delhi 22 Ranjan Bose Department of Electrical Engineering

So, let us see how we can use it, because finally our aim is to decode a received vector. Suppose this is the codeword in use, and what we receive at the receiving end is 1 1 0 1. Now, clearly 1 1 0 1 does not happen to be any one of the valid codewords, so it rings a bell, and detection is almost immediate right. So, we deduce that clearly an error has happened. If it had matched, we do not go to the next step, we declare the result. Now, which now we try to estimate, which one of the four possible codeword was actually transmitted, that is the decoding problem.

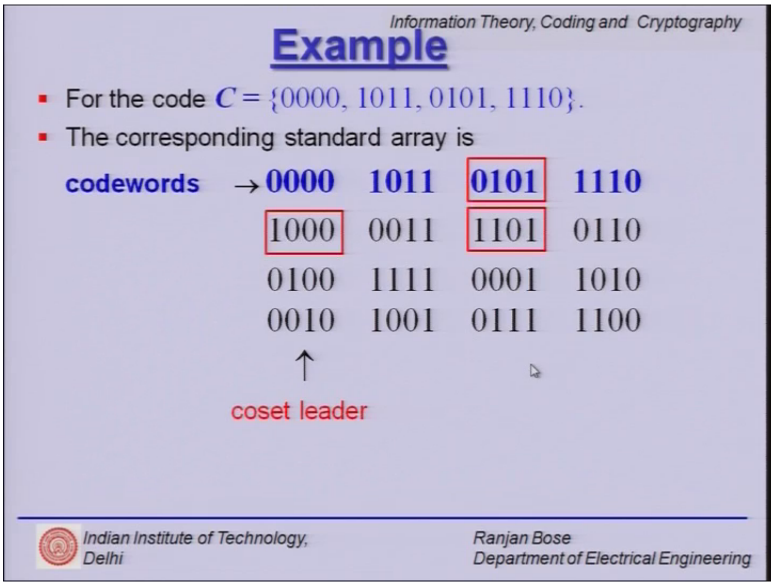
So, we use make use of the standard array in the earlier example, and find that 1 1 0 1 lies in the 3rd column. And the topmost entry of this column is 0 1 0 1. And hence, the estimated codeword is 0 1 0 1.

So, basically if you go back, and you have received this vector so, you have this nice array, this standard array available with you. And you are looking for, which of them has come. Please note all the sixteen code words are listed, so whatever you receive, any vector of length four has to be one of them. We happen to receive 1 1 0 1. So, clearly, we quickly say that we move up, and see that look this will be the most likely codeword that was to be sent. And the error will be 1 0 0 0 this is the error vector.

So, what has happened is, the first bit is an error it has gotten flipped, we really sent out 0 1 0 1, and near by decoding cell says that 1 1 0 1 was received. And you can verify that this is closer to this than any other codeword as per the standard array decoding.

Now, how do we check, you can check this distance with all other valid codewords. And you will see that indeed 0 1 0 1 this codeword is closest to the received vector. And hence, you have been able to decode it correctly. And the coset leader automatically becomes the error vector. And what we do for error correction is just add the error vector one more time and the correction happens. So, that is how you use the error vector.

(Refer Slide Time: 33:53)



Information Theory, Coding and Cryptography

Example

- For the code $C = \{0000, 1011, 0101, 1110\}$.
- The corresponding standard array is

codewords →	0000	1011	0101	1110
	1000	0011	1101	0110
	0100	1111	0001	1010
	0010	1001	0111	1100

↑
coset leader

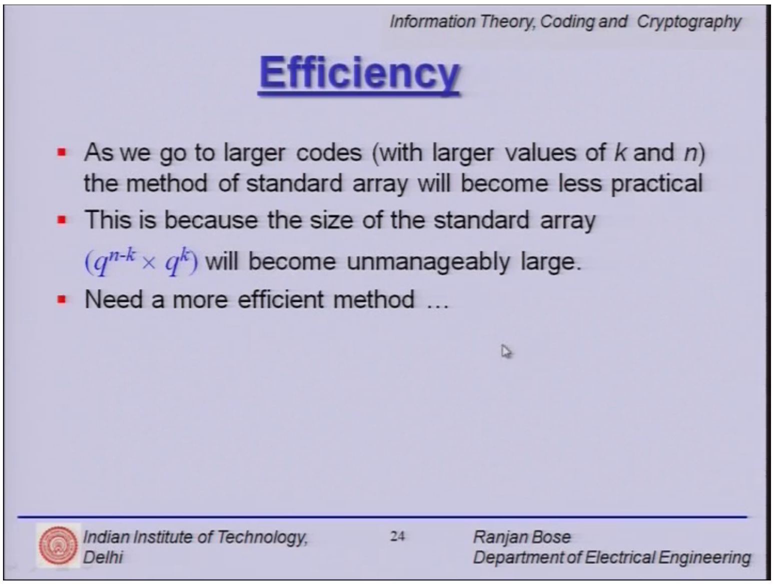
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

So, this is the thing we had realized 1 1 0 1 was received, and immediately I look-up, because it these are the four usual suspects. And immediately I look-up in the same column and 0 1 0 1 is there, and on the left is the coset leader, which is the error.

So, very efficiently and quickly we can come to not only the error detection, but error correction part using a standard array. Of course, you can see that as the size of the codebook increases. As we have more and more number of codewords in my table, this array will grow exponentially. So, there will be a problem with large n and k.

(Refer Slide Time: 34:44)



Information Theory, Coding and Cryptography

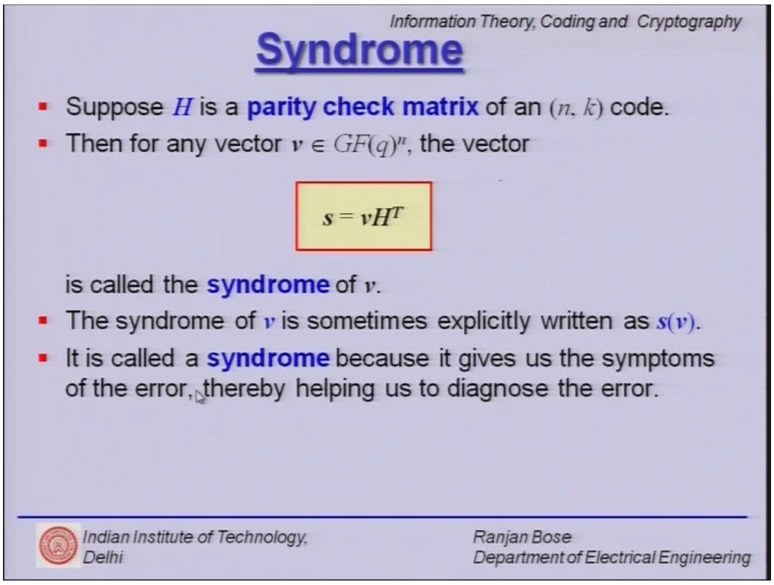
Efficiency

- As we go to larger codes (with larger values of k and n) the method of standard array will become less practical
- This is because the size of the standard array $(q^{n-k} \times q^{k})$ will become unmanageably large.
- Need a more efficient method …

Indian Institute of Technology, Delhi 24 Ranjan Bose Department of Electrical Engineering

So, as we go to larger codes, with larger values of k and n, the standard array technique becomes less practical; Nonetheless, if it in principle always holds, and is used to proving certain theorems also. But, in general for practical perspective, we will not be using standard arrays, because this table will be q raise power n minus k cross q raise power k. So, clearly we need a more efficient technique, do we have one, the answer is yes, and it comes in the form of syndrome decoding.

(Refer Slide Time: 35:16)



Information Theory, Coding and Cryptography

Syndrome

- Suppose H is a **parity check matrix** of an (n, k) code.
- Then for any vector $v \in GF(q)^{n}$, the vector

$$\boxed{s = vH^{T}}$$

is called the **syndrome** of v.

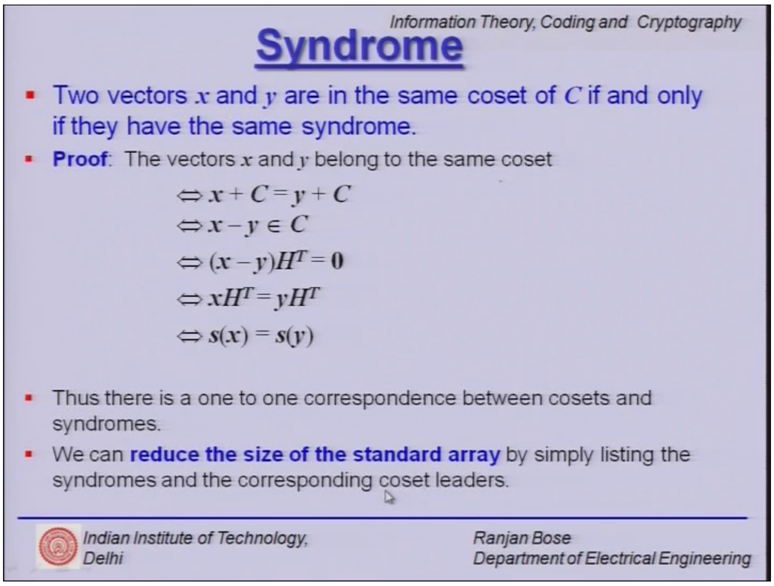
- The syndrome of v is sometimes explicitly written as $s(v)$.
- It is called a **syndrome** because it gives us the symptoms of the error, thereby helping us to diagnose the error.

Indian Institute of Technology, Delhi Ranjan Bose Department of Electrical Engineering

So, let us first look at what do we mean by a syndrome. So, we have this parity check matrix H, which so far was doing the first job of error detection. And error detection means, I have v H transpose if it is equal to 0, it means that error has not occurred, and if it is a non-zero vector some error has happened ok. The question is which error. So, we define this syndrome s as the received vector nu times H transpose. Clearly, in this received vector nu is the valid codeword, then it becomes v H transpose, and s becomes 0.

But, in the even that nu is not the valid codeword, it is valid codeword plus some error. Then s will no longer be the 0 vector, then what will it be, it will be called the syndrome. So, s is called the syndrome of nu. Syndrome is sometimes explicitly written as s as a function of nu. Why is it called a syndrome, it is called a syndrome, because it gives us the same symptoms of the error, thereby helping us to diagnose the error.

(Refer Slide Time: 36:57)



Information Theory, Coding and Cryptography

Syndrome

- Two vectors x and y are in the same coset of C if and only if they have the same syndrome.
- **Proof**: The vectors x and y belong to the same coset

$$\Leftrightarrow x + C = y + C$$
$$\Leftrightarrow x - y \in C$$
$$\Leftrightarrow (x - y)H^T = \mathbf{0}$$
$$\Leftrightarrow xH^T = yH^T$$
$$\Leftrightarrow s(x) = s(y)$$

- Thus there is a one to one correspondence between cosets and syndromes.
- We can **reduce the size of the standard array** by simply listing the syndromes and the corresponding coset leaders.

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

Suppose two vectors of x and y are in the same coset of C right. So, two vectors x and y in the same coset of C, if and only if they are having the same syndrome; This is very interesting, because it tells us that the concept of coset based standard array that we came up with last time, which logically partitioned your space into clearly decodable subsections. We have now the same syndromes for the same coset.

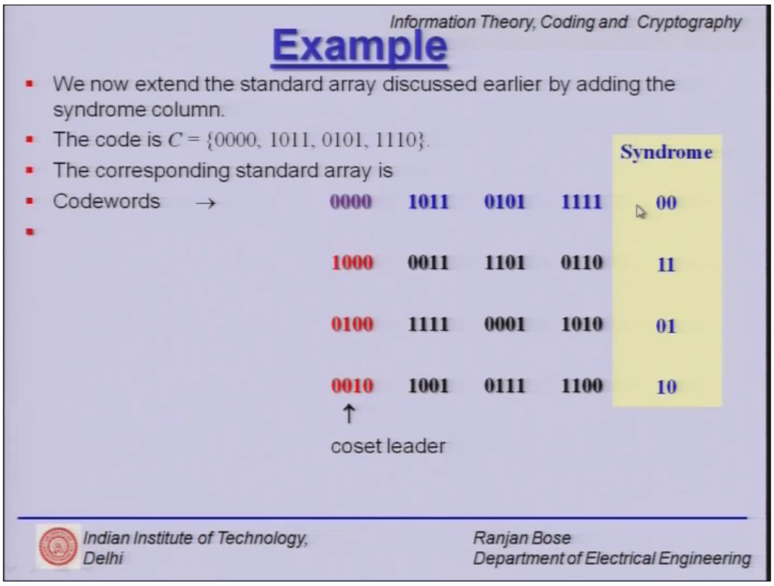
And please remember, the cosets were having the same error, with the coset has a coset leader and that is the error. So, this is a relief, because if you have the syndrome corresponding to the error and not the information word, then we are in business. Just like a doctor looks at the symptoms, and gives out the outcome of a possible disease, it does not matter, which patient has that. So, the symptoms tell you about the disease, but since the symptom together with the patient does not have to be used.

Similarly, the error vector should itself directly lead to the syndrome, it does not matter, which codeword is an error. So, the syndrome is only corresponding to that particular error, this is a big revelation, this is a big observation, which helps us efficiently do things.

So, the vectors x and y belong to the same coset so, these are the translates x plus C is y plus C and x minus C is an element of C. So, x minus y H transpose is 0 and therefore, x H transpose is y H transpose that is they have the same syndrome. Thus there is a one to one corresponds between correspondence between cosets and syndromes. This relieves us, because we can reduce the size of a standard array by simply listing the syndromes, and the corresponding coset leaders. And who are these coset leaders, they are the error vectors.

So, we just need to have the a table, which has syndromes and coset leaders, a table of syndromes and errors. So, if you have a syndrome, we can directly control, which error it is ok. So, this really simplifies our life.

(Refer Slide Time: 39:56)



Information Theory, Coding and Cryptography

Example

- We now extend the standard array discussed earlier by adding the syndrome column.
- The code is $C = \{0000, 1011, 0101, 1110\}$.
- The corresponding standard array is
- Codewords →
-

				Syndrome
0000	1011	0101	1111	00
1000	0011	1101	0110	11
0100	1111	0001	1010	01
0010	1001	0111	1100	10

↑
coset leader

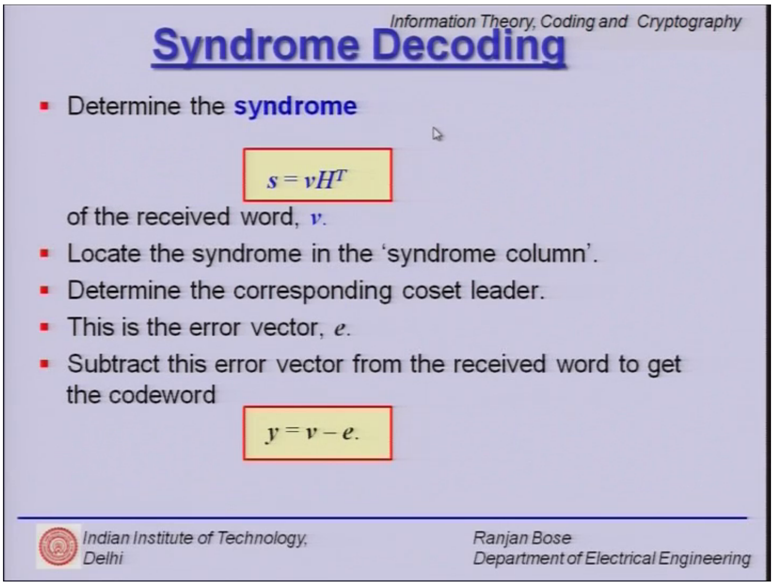
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

So, we now improve upon a standard array by adding a syndrome column. So, this is our code C, and we have already looked at the corresponding standard array, and I have put down in red the coset leaders, and in blue we have the code codewords. But, now this is the coset leader, and this is the error that will be caused and leading to these corresponding cosets.

Now, what you do is put a corresponding syndrome table out here. And for every coset leader you have a syndrome. So, the syndrome vector is 1 1, you clearly know that this is the error. If the syndrome is 0 1, you clearly know that this is the error. You do not have to bother about, which codeword was sent. And clearly the syndrome is 0 0 for the valid codewords.

(Refer Slide Time: 41:08)



Information Theory, Coding and Cryptography

Syndrome Decoding

- Determine the **syndrome**

$$\boxed{s = vH^T}$$

of the received word, v.
- Locate the syndrome in the 'syndrome column'.
- Determine the corresponding coset leader.
- This is the error vector, e.
- Subtract this error vector from the received word to get the codeword

$$\boxed{y = v - e.}$$

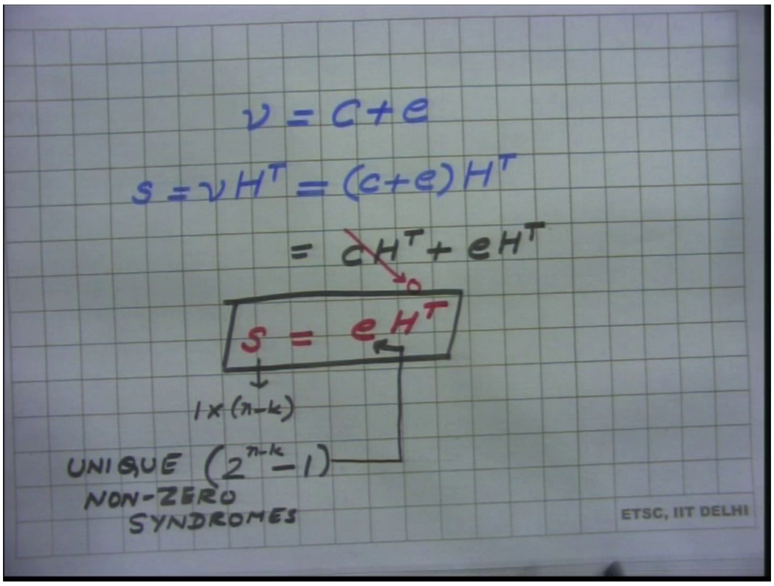
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

So, now we come to this notion of syndrome decoding. What do we do for this, we first determine the syndrome. And how do we do that, we always get a received vector nu, we always have an H matrix, which is the parity check matrix, we calculate nu H transpose. If you are lucky, we get a 0 vector, and we said that look we have no errors. In the case, that we do not get a 0 vector, we get the syndrome vector s.

Now, this syndrome vector, the syndrome is located in that syndrome column of a standard array, and we directly move left to determine the corresponding coset leader. Who is this coset leader, this is the cause of error, this is the error vector ok. This automatically is your error vector, e. And subtract this error vector from the received word to get the valid codeword, the codeword that was sent, and that is the end of syndrome decoding y is nu minus e. So, these are the steps, which tell us how to do syndrome based decoding.

(Refer Slide Time: 42:42)



$$\nu = C + e$$

$$s = \nu H^T = (c+e)H^T$$

$$= \cancel{c}H^T + eH^T \quad 0$$

$$\boxed{s = eH^T}$$

$$1 \times (n-k)$$

UNIQUE $(2^{n-k} - 1)$ NON-ZERO SYNDROMES

ETSC, IIT DELHI

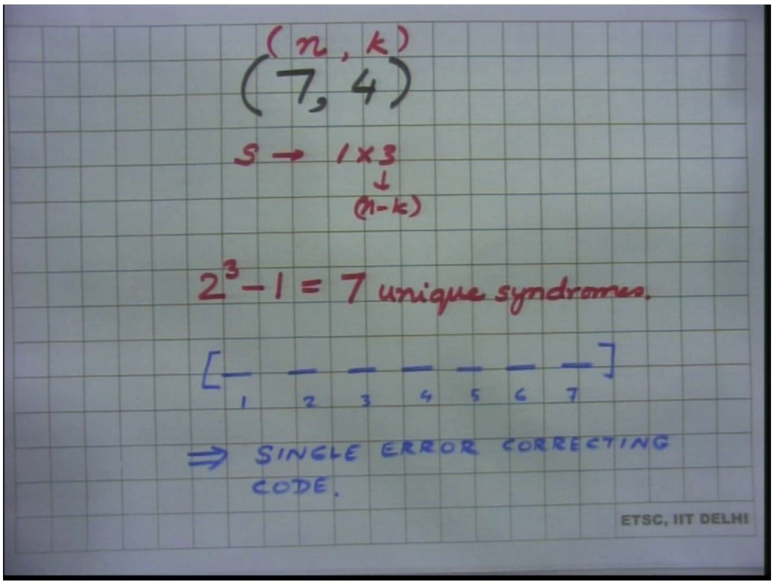
So, before we conclude let us look at a very interesting observation. Suppose we have a code word C, which was sent. But, on the weight encountered in error e, and what we received was nu. At the receiver, we only have nu to work with. So, what we do is we take this nu, and multiply with is H transpose, and this is what we called a syndrome. But, this nu is nothing but my actual codeword C plus my error vector e, and they get multiplied with my H transpose.

Now, this can easily be split out, and you can write it as C H transpose plus e H transpose. But, we know that C H transpose by definition is 0, because the job of this parity check matrix H is to make sure that C any valid codeword C, where multiplied with H transpose gives a 0, so this is a 0. So, what we obtain is e H transpose. So, your syndrome is nothing but e H transpose.

Now, we so to say killed two birds with one stone. First we make sure that s is not a function of C right. The symptom is not dependent upon the person, who is reporting the disease. The symptom is only a function of the disease the error. So, the syndrome depends only on the error, and clearly not on the codeword, this is an important observation. Now, please note that this syndrome will be of the skies 1 cross n minus k.

So, we have if in the case of binary, we have only 2 raise power n minus k possible unique syndromes, of which one of them is all 0 vector. So, this is the total number of unique non-zero syndromes right. Now, these must indicate, which error has happened. So, it can only handle so many errors all right.

(Refer Slide Time: 46:01)

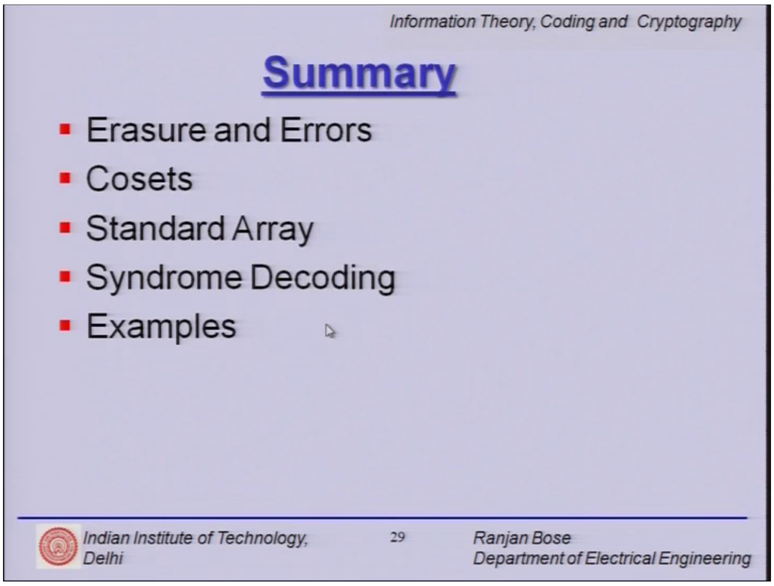


So, for example let us understand it by a simple example. Suppose we have an 7 comma 4 code. So, here your n is 7, and k is 4. If you wish to be syndrome decoding, then your syndrome will be of length n minus k. So, your syndrome is of length 1 plus 3, because n minus k is 3, which means that 2 raise power 3 total number minus the all 0 vector is equal to 7 unique syndromes are available.

Now, we make a quick observation we observe that the 7 comma 4 code is always a vector of length 7. So, location 1, 2, 3, 4, 5, 6, 7, each time I receive a vector, it is of length 7. Now, if I wonder, how many errors can my syndrome decoding correct, my answer comes right in front of me, because if there is 7 unique syndromes possible, and each one should correspond to one error, then this errors would basically be error at location 1, error at location 2, 3 and so and so forth. So, each one of these unique syndromes would actually tell me, which is the error location, and that is it.

This code the 7 comma 4 code can at best be a single error correcting code at best. If I am ambitious, and if I try to say look can I squeeze in 2 bits to be corrected, there is no way, because there is no freedom. I do not have that much of unique syndromes available for me to assign to possible error corrections in this case. We at best have 7 errors, and they correspondent to correspond to 7 locations. So, the 7 comma 4 code is a single error correcting code, we cannot have anything better than this. So, it is a general way to look at how things work, and what is the freedom and flexibility we have in terms of syndrome decoding all right.

(Refer Slide Time: 49:12)



Information Theory, Coding and Cryptography

Summary

- Erasure and Errors
- Cosets
- Standard Array
- Syndrome Decoding
- Examples

Indian Institute of Technology, Delhi 29 Ranjan Bose Department of Electrical Engineering

So, we now come back to our slides, and we summarize what we have covered so far. We started off with erasures and errors; we looked at channels, which can introduce both errors and erasures. And what detrimental effect it has on the minimum distance d star. And what is the number of errors we can correct given are erasures occur.

Then we looked at a very important concept on regarding cosets. And we learnt how to form standard arrays using cosets. And then, we looked at a more efficient way of decoding called syndrome decoding. Finally, we considered some examples. With that we come to the end of this lecture.