COMP28512

Lab\_Task 5 Report

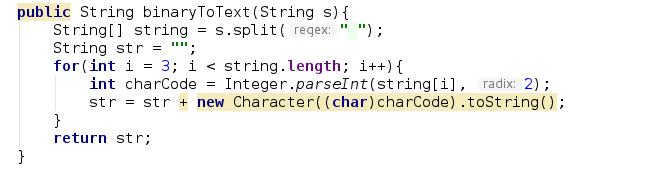
Question 1.1: Explain and demonstrate the code for the conversion and re-conversion.

When I use the <mode>MSG function to simulate the error. I need to change the string to bit. After I receive the ASCII type message from barrybot, I translate to string.

As the picture shown below, I send a command “**2MSG BarryBot hello manchester**”

the response of the barry bot simulate the bit error.





Question 1.2: How did you enter the message and did you get back the same message exactly?

I enter the message in the form “<mode>MSG BarryBot <text>”

As the picture shown below, I send a command “**2MSG BarryBot hello manchester**”

the response of the barry bot simulate the bit error.



But if I type “**0MSG BarryBot hello manchester**”, It will get back exactly the same

message :”**hello manchester**”

Question 1.3: Although the simulated channel should not deliberately introduce any bit-errors in mode 0, what would happen if a real bit error occurred due, perhaps, to a malfunction of the mobile phone? How would your program deal with this occurrence?

I can use CRC to develop the function to correct the error

Question 1.4: How does this form of transmission increase the required bit-rate over the simulated channel?

If I want to send a character, I need to transform to 8 byte 01001010, so it will be 8 times bit rate.

Question 1.5: Why is this form of transmission useful for experimental purposes?

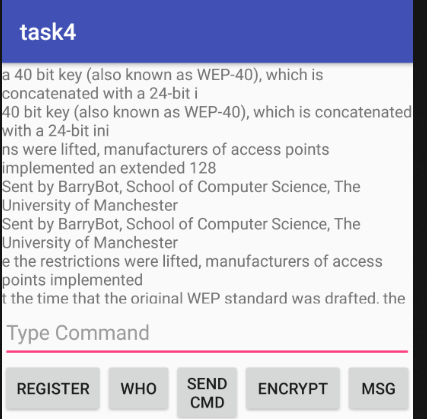
Because it is easier to simulate bit error in this form. It is more likely to create errror.

Question 2.1: What are your brief observations?

The error rate for 1 2 3 is 1<2<3. it is most likely to find error in mode 3. less likely for mode 1.

Question 2.2: How realistic is it to assume that bit-errors will normally be evenly spread out in time?

It is like a real bit error that will happen in real mobile information transfer. There are more errors in mode 3. and less error in mode 1.

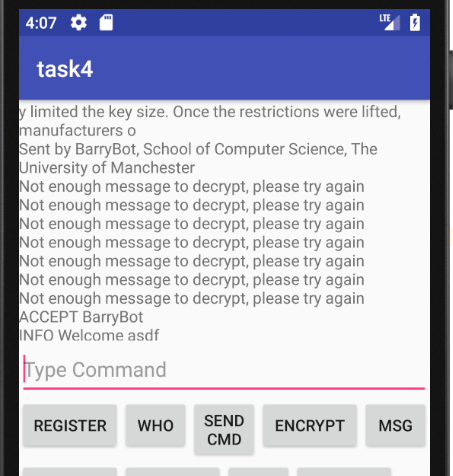
Question 3.1: What were the messages?

Here is the message after I decrypt the barrybot.

Question 3.2: Assuming that this experiment demonstrates that using the same key more than once is dangerous, how could you avoid this danger while still allowing the receiver to deencrypt your messages?

create a new key over a period and change the key periodically. OR every time I start the barrybot, new a key.

Question 3.3: Briefly summarize your de-encryption method and indicate whether it really worked.

 When I havent decrypt successfully, I will get message “not enough message to decrypt.”After I decrypt it, the message will be received.

For decrypt method. The logic is firstly find the encrypted signature in ASCII form. Then XOR with the know signature in ASCII form. To get the key. Then use xor to decrypt all the rest information.



Question 3.4: If you could not predict that an email signature will be sent regularly, what other weaknesses could make the encryption vulnerable to attack.

The weakness is the key is not changed.

Question 4.1: What happens if you keep pressing the ‘start’ button?

The music will continuously play for the time I clicked. If I click for 3 times, the sound will play for 3 times.

Question 4.2: Why does a ‘wav’ file have a 44 byte header, and how are the sound samples themselves stored? Are the samples stored in text form?

It is used to define the size of the wave file.

Question 5.1: How successful is your simple method of dealing with lost packets in speech transmissions?

I add logcat in my method so that I can know when the packets are lost in speech.

Question 5.2: How successful is your simple method of dealing with bit-errors in speech transmissions?

I use logcat to locate the packet replacement.

Question 6.1: How did you display the original undamaged images?

I use imageView to display the image

Question 6.2: What was the effect of the bit-errors on the bit-map image?

Some “noise” pixel will appear on the picture. As the bit error happen, not every pixel is in their correct colour.

Question 6.3: How could you conceal the distortion caused by these bit-errors (if you had time)?

I can find a proper colour for the pixel that affected by bit error. By Calculate the colour value of it from the pixel near it.

Question 6.4: What was the effect of the lost packets on the bit-map image?

There will be a black line appear. Because the image is a array, and lost packets are loss of continually index in this array.

Question 6.5: How could you conceal the distortion caused by these lost packets (if you had time)?

I can replace the black area with the same length of image array index above.

Question 6.6: What was the effect of the channel errors on the JPEG image? Question 6.7: Why are ‘jpg’ images much more sensitive to bit-errors than ‘bmp’ images?

Because jpg file is after compressed. So the packts are more important because they are responsible for mor pixels.

Questions:

7.1. What was, or could be, the effect of the design flaw.

It will cause noise in background.

7.2. How did you correct the design flaw?

I can get the array form of the sound and to smooth it

7.3. Demonstrate the effect of the bit-errors and packet-loss in modes 0 to 5.

as the mode get higher, the more sound is lost.

For bit error, there are more noise, for packet loss, the sound get lost.

7.4. What was the effect of increasing the sampling rate to 12 kHz on play-back?

If increase the sampling rate, the sound is more clear and less effect by bit error.

7.5. Why is it useful to be able to develop your own application for speech recording or processing, rather than relying on downloaded apps. Suggest a useful application that you now know how to develop.

Because I can learn a lot from the process when I develop the application. There will be many difficulty, like deal with bit errors or packet loss. I know how to develop a useful application that can record the sound.