Transcription of “The Magic Picture" Lesson

First Segment:

Once upon a time, a woman came to a ruler and said," I complain to you that I do not have mice at my place!" People around the ruler started wondering, but he understood what she wanted to say, and ordered, her home be filled with bread, meat, ghee and dates.

Hello everybody (Assalamu Alaykom), and welcome in this lesson.

My name is Abdullah Saleh Seddiq. I am a computer programmer. I work here at Prince Sultan bin Abdul Aziz Science and Technology Center (Scitech) in Khobar, KSA.

We are going to talk about ciphering, which is basically converting data from a form understandable to everyone to another form that is understandable only by people who have a specific knowledge. Ciphering has been used in old ages in military and diplomatic letters, also in espionage.

Arab-Muslim scientist, Yacoub bin Ishaq Al-Kindi, who lived in Baghdad during Caliph Al-Mamoon's reign, is considered the first to put a method for deciphering which is documented in book "The Science of Finding out Ciphered Texts "

Today's lesson is about ciphering and steganography by using the computer, which is one of the most important tools in our modern era.

Do you think it's possible to cipher and hide a computer file that I want to send inside a digital picture, with no change whatsoever to in its shape or size? And why?

Give it a quick though and discuss with your colleagues, and I'll see you in two minutes.

Second Segment:

Hi, perhaps many of you said that it is not possible to hide a file inside a picture, but let me tell you that it is feasible by ciphering and hiding (Steganography)

First, let us mention that a computer is a device which purely relies on ciphering (coding); but it is not a secret ciphering. In fact, it is known by all programmers. The Computer is an electronic device which can only recognize two statuses, and this is the fundamental base of the binary system.

Specific voltage is considered logic 1; another smaller voltage is considered logic 0.

Letters, numbers and other basic symbols which people understand and use daily have been coded using combinations of zeros and ones, and each 8 is called byte, which is

a measurement unit of a file size in computers, as you know. A bit is one part of this byte (8 bits). The American Standard Code for Information Interchange (ASCII) has adopted this methodology, which contains all these characters – a total of 256 characters.

Note that (capital "A") is character 65 in the table, while (small "a") is character 97, ("0") is character 48, and (space) which separates between words, is character 32, and there is a group of math operations and other symbols.

Example: The word "Salaam" will be coded as follows:

"S": (83) = [01010011]

Let's recall how to convert from decimal to binary system, where each bit has a given value. We need to select a group of numbers of a sum of 83, and put 1 in the bit which is over each of them. Then we put zeros in the remaining bits as follows: 1 1 1 1 and the others are zeros.

In the same way, "a": (97) = [01100001]

"l": (108) = [01101100]

"m": (109) = [01101101]

Now, let us find a method to cipher a sentence of few words using our own ways. We are going to divide you into three groups:

Group (A) will write a ciphered sentence, and give it to group (B)

Group (B) will try to decipher it through guessing, testing and trying. If they could not, we will give it to group (C), which would decipher using the method which their colleagues in group (A) would give.

Do this exercise in cooperation with your teacher, and we will be back in few minutes.

Third Segment:

Welcome back. I am glad you kept trying, and I am really impressed with those who were able to decipher, and as you have noticed it is one of the easiest methods to replace each letter with the next letter in the alphabet, or the previous one, or use our own designed table for replacing letters.

You have found that it is possible to decipher through guessing, or by noticing the repeating pattern of some letters which is a method used first by scientist Al-Kindi (Frequency Analysis).

Don't you agree that ciphering alone is not enough? It perhaps triggers some people to prevent a message to be delivered out of curiosity, or make them try to decipher it to

discover its contents. Don't you think it's good to do some hiding?! Let's try this. We have a sack here.

We have something inside the sack. Can you tell what it is? It is a ball. We can guess that without seeing it. As you see, this is not a suitable hiding.

Let's try something better. We have here two glasses of flavored tea. I am going to put some sugar inside one of them, and stir both glasses, and then I shall ask my friend Abdul Kareem this question: "Excuse me, Abdul Kareem. Which one of these two glasses contains sugar? Can you tell by seeing only?"

Abdul Kareem: “No, I can't."

Thank you, Abdul Kareem

Now, let’s find something like that. Let’s see how Paint in Windows stores 24-bit bitmaps. It is like this panel.

Here we have a 4 pixel picture enlarged 2500 times. I am going to store this bitmap, and re-open in a hexadecimal editor, known by programmers.

As you see, the picture is encoded by this information. We have here a BMP file. Note that there is a header of 54 bytes, and bytes from 35 to 38 define data size. After header, there is data, which encodes colors used in coloring each pixel. There are 3 bytes for the color of each pixel, starting from the left pixel in the last line of the picture, and ending by the right pixel in the first line of the picture. Where:

First byte is for the amount of blue color in the pixel.

Second byte is for the amount of green color.

Third byte is for the amount of red color.

Each value belongs to the range [0 to 255].

And that is why we can select the color of a pixel, one of about 16 million colors, because 256 X 256 X 256 = 16777216 colors, which is the probability of the color of a pixel.

Also we notice that there are "extra bytes" which have value of 0, used to make the number of bytes, which represent a line of the picture, multiple to 4. This is according to the design of 24-bit BMP format.

May you think about other formats like jpg? Yes, we can use it, but it's a little bit hard.

And now, while I am drinking this glass of tea, try –each group alone- to find a method (to dissolve), sorry, to cipher files in pictures. See you soon.

Fourth Segment

Have you found a way? Well, maybe you have noticed that our eyes are not able to distinguish 16 million colors like the computer. We can derive benefits from that point. Imagine that we have 256 colors gradual from white to dark red: white, red, more, more, more, dark red. Can you distinguish between red 255 and red 254? I do not think so.

Now, we are closer to achieve our goal, which is to cipher and hide a file inside a picture or photo, without a noticeable change in shape, or size.

Try to find a place to store files inside bitmaps.

Fifth Segment

Thanks for your answers. Of course, we can use the lowest bit in pixels' bytes, to store bits of file which we want to cipher, thus all files' bits are replacing lowest bits in the bytes of the colors without affecting the picture itself. Remember that all files such as texts, pictures or any other types are a set of bytes.

Also, we can use 2 bits, or 3 bits, or maybe 4 bits, of course lowest bits, to store a ciphered file. That will cause changing about 15/255, which is not more than 6% of the color quality, and you will see after a few minutes that these changes sometimes cannot be recognized easily by our naked eyes in high quality pictures with a rich variety of colors.

Now, I want you to answer the following questions in the break:

1- Calculate the size of hidden data, which a picture can have in each of the following cases:
   a- Using 1 bit.
   b- Using 2 bits.
   c- Using 4 bits.
2- What happens if we used 8 bits?

Sixth Segment

Let's calculate the size, keeping in minds that we need to add extra bytes for a word to show that the ciphered file has finished. We need that while deciphering to know that we reached the end of the file since there is no harm in having a picture bigger in size than what the ciphered file needs.

For example, the word (FSNSTNBKTK#) has no meaning! And since we assume that it cannot occur in any file we want to cipher, then we can use it as a sign that ciphering has finished. Its letters are 11, so we need 88 additional bits.

If the file to be ciphered is 1 MB, which is 1024 kilo bytes, which is 1048576 bytes, which is 8388608 bits. We add 88, and then it becomes 8388696.

Using 1 bit only in each byte in each pixel in the picture, we need 8388696, and because each 3 bytes form a pixel, we need 2796232 pixels.

By square rooting this number, we find that we need a squared picture of 1673 * 1673 pixels. But it is recommended to use a picture of 1676 * 1676.

Because if we have 1676 pixels in the line of the picture, and each pixel is stored in 3 bytes, we will have 5028 bytes, and this number is a multiple of 4, and that means there are no extra bytes in the picture that can lead to revealing the hidden data, so these bytes must have the value of 0 only.

So to cipher and hide a file: text, table, presentation or any other type, with a size of 1 MB, we can use a bmp picture of 1676*1676 pixels, its size is about 15 X 15 cm. It is like photos captured by cell phones or digital cameras with a resolution power of 3 Megapixels. We can summarize that in this relation:

$x>=8*(s+n)/ (3*b)$, where:

x is an integer which represents the number of pixels in the picture (Width X Height)

s is the size in bytes of the file to be ciphered

n is the number of bytes for the end- sign word (FSNSTNBKTK#)

b is the number of low bits, we want to use, in each byte in each pixel of the picture.

Using this relation we find that:

Using 2 bits, we will need 1398116 pixels

Using 4 bits, we will need 699058 pixels (Which is a 7 X 7 cm picture with 300 dpi BMP)

That is for ciphering 1 MB file, which is the size of a large book of 800 pages.

If we used 8 bits, the picture will be changed 100%, so it will be obvious that it contains ciphered data.

Now, I will let you think for few minutes about this question:

What are the requirements of writing a computer application that can do this ciphering and hiding?

Seventh Segment

Welcome again. Really, I am sure that most of you, who have some knowledge in programming, and know how to write conditional statements and repetition loops, are able to write an application to do this ciphering. All what you have to do is the following:

1- Open and read a byte from a binary file (The picture + the file to be ciphered)
2- Divide a byte to bits, and put them in the bytes of the picture
3- Add end-sign word, and save the resulted binary file (The magic picture)

For me, I have chosen to write a computer program using Visual Basic to do this ciphering and deciphering.

You can download it from BLOSSOMS web site.

I will cipher this small simple text file, inside my personal photo.

Look how much it is easy. We run the program and select "Put".

We select the picture and the text file, and set the path and name for the resulted file, then press Start

Wait a moment, and everything is done.

Now, look at the resulted magic picture, it seems identical to my personal photo.

Its size is exactly the same size of my personal photo. Wonderful thing!

Let's cipher this magic personal photo and hide it inside Scitech picture.

We repeat the same steps, compare and see that everything is alright.

Now, Let's delete these files which we used, except the magic picture of Scitech.(The last one.)

Let's run the program again, and decipher by using “Get"

Look. Ha, my personal photo is here again, and it is a magic one as well.

Let's decipher it by using "Get"

Oh! The text file is here again. Let's open it and check its contents.

Really, it is a wonderful application, isn't it?

You may also test ciphering files and hide them inside photos of your cell phones, after changing their format to 24-bit Bmp, using "Paint" in Accessories of Windows system.

Programming is very interesting. I wish that you do your best to learn it, and write your own applications.

God bless you, and good bye.

Eighth Segment (Teacher's Guide)

Dear Teachers

Assalamu Alaykom (Peace be upon you)

As you have seen, this lesson aims to relate many topics in computer science in order to have a useful application in real life.

The student will get to know the concept of ciphering and hiding (Steganography),

In addition to an overview of the American Standard Code for Information Interchange table "ASCII" , the structure of bitmap (BMP), and how colors are stored.

Also, the student will learn how to do calculations involving byte and bit, and how to manipulate binary files .Eventually; the student will be able to use computer applications using Visual Basic, or other programming languages to do ciphering and hiding a file inside a picture, without a noticeable change in shape or size.

It is highly recommended that the students have a previous knowledge of:

1- Binary System.
2- Computer storage unit "Byte" and its multiples: Kilo, Mega and Giga.
3- Hexadecimal Editor.
4- Programming basics, and manipulating binary files

At the end of the second segment, dear teachers, you may help students by suggesting methods for ciphering like: Changing letters order, or replacing a letter by another according to their own table, or to use the previous or next letter in alphabet.

Your cooperation and knowledge in the computer field are required as well in any other questions raised by your students.

Finally, there are many points which you can discuss with your students, specially the gifted ones, or those who are interested in this topic, like:

- Can you cipher and hide the file inside more than one picture? (Distribute it in more than one picture.).
- What about writing an application, using one of programming languages, and making it camouflaged where it appears to be doing another purpose other than hiding and ciphering?

- How can you do some modifications on the application to make it suitable for many groups of users, so that any group cannot decipher magic pictures for other groups? (Through using shifting option, or different ciphering tables)
- Allow user to select the number of low bits in the picture, which he wants to use in ciphering and hiding (So that the user controls the unnoticeable small changes in the picture)
- Are there any ways to reveal the contents of the magic pictures without using the application itself? And how much time will be needed to try the probabilities if any?

At the end, I wish that BLOSSOMS team and I have succeeded in adding a valuable lesson to the scientific library.

Assalamu Alaykom (Peace be upon you)