

Teacher's Guide: The Magic Picture

Abdullah Seddiq, Scitech – KFUPM – KSA

Thank you for considering The Magic Picture for your class. I hope you and your students enjoy the experience. This short teacher's guide is written as a supplement to the video Teacher's Guide included at the end of the video segments. So, if you have not yet viewed that, it would be best to view it now, and then return here! Thanks!

I think that this lesson is good also for those who are not interested in IT or programming, because it gives an idea about steganography in easy and simplified way.

Let me mention following points;

- Al-Kindi: I have mentioned in first segment that Al-Kindi set a method for deciphering. It is called "Frequency Analysis" which is still used until now. It tells that one way to solve an encrypted message, if we know its language, is to find a different plaintext of the same language long enough to fill one sheet or so, and then we count the occurrences of each letter. We call the most frequently occurring letter the 'first', the next most occurring letter the 'second', the following most occurring the 'third', and so on, until we account for all the different letters in the plaintext sample.... Then we look at the cipher text we want to solve and we also classify its symbols. We find the most occurring symbol and change it to the form of the 'first' letter of the plaintext sample, the next most occurring symbol is changed to the form of the 'second' letter, and so on, until we account for all symbols of the cryptogram we want to solve. (Al-Kindi in his 9th-century A Manuscript on Deciphering Cryptographic Messages)
- ASCII: You will find a copy of ASCII table with this lesson. We have to mention that some characters in this table is for control. They cannot be printed as text.
- Convert from binary to decimal: Sum the values of bits which contain 1 and ignore those who contain 0. It is preferred to cipher characters before hiding them in the magic picture. So if you have 'A' convert it to 'G' for example. You can do that through special table you write it for that purpose, or by adding certain number to the byte value of that letter using Ord or Asc and Chr functions in programming languages.
- Hexadecimal Editor: It is an application you can use it to open any computer file, to see its bytes in hexadecimal value or as text. This program is used by programmers and hackers so they can change many things in files. I used AXE editor, it is free ware, and you will find it with files of this lesson.
- Colors: You may use Paint in Windows to show your students how colors are made by mixing 3 amounts of red, green and blue.
- Extra Bytes in BMP: If we have, for example, a bitmap with 29 pixel width, so bytes which represent a line of that bitmap are $29*3=87$ which is not multiple of 4, so Paint add 1 extra byte to make it 88 which is multiple of 4. (Sometimes 3 extra bytes is add to the bytes of each line in the picture to make them multiple of 4)

- Activity 4: Students may suggest to store data in the extra bytes, here we have to mention that they may not exist in some pictures, or their amount is very small which is not enough to do data hiding. And some photo experts may discover that there are hidden data, if these extra bytes did not have 0 value, which they have to have.
- The original picture: It is important to clear that after hiding data, the magic picture alone is enough for the process of deciphering, and it is impossible to retrieve the original picture again, and that is not important, and if you want to be more secure, you have to delete the original picture to prevent others' comparison between the two pictures (Original & Magic)
- The magic picture has the same size of the original one exactly, and its shape SEEMS the same but it is not the same exactly. If we used 4 bits then the human eye may see the difference hard, but if we used 1 bit then it is impossible to see any difference without using a computer. Try to hide a file in an empty (white) picture, then try to fill with red the resulting magic picture, you will find that there are some areas that cannot be filled because each pixel is different from others around it.
- Time: We prepared this lesson in such a way that you can do it in your 45-50 min class. But if you want to speak in depth about the issues mentioned in the lesson and write with your students an application using a programming language then you may need more time. (The application will be about 100 lines of code)
- Contact me: Finally, Please don't hesitate in sending any questions, suggestions or comments you have, to my email: abdullah@salaamsoft.com

Good-bye