Linnaeus University

1DV700 - Computer Security
Assignment 1

Student: Fredric Eriksson
Personal number: 001124-1239
Student ID: fe222pa@student.lnu.se

Setup Premises

- Windows OS .
- HxD Hexeditor.

Task 1

A.1) The difference between both methods are that in the symmetrical there is only one key, hence its very safe due to only User A and User B know the key, the only issue is that it might be troublesome to send that key from User A to User B. In the other method the keys are produced together or one is derived mathematically from the other. Thus, a process computes both keys as a set.[1]

A.2)The difference between both are Encryption algorithms are used to send private data from User A to User B, its very personal and its usually hard to decode because a PC might take too long finding the for instance the phi value sent from User A to User B, this method of sending information requires high amount of coordination from both users which might make the method quite difficult to execute.

the second method is a manner to identify certain documents, for instance a document file might have a determined amount bites and text, after hashing said document a unique value is created, that value is a way for everyone to recognize the document so it is valid.

The issue with this method is that if anyone else manages to create a similar document with the exact same amount of values then the documents could be changed and the introduced document might be considered the original in which might lead to confusion or catastrophe.

A.3)The difference between both are: Compression is about making a file as small as possible by changing many of the fundamentals of the file so it occupies less space, hashing just provides a type of “Serial code” that allows all users and computers to detect the file and determine if it is the original one.

B)Stenography hides messages inside a picture, mostly from the hex code, because changing a hexadecimal number for a color it is not recognizable for the human eye, it can be used at best to send messages between 2 entities or at worst case a way to trojan horse into a pc in order to execute files. Encryption changes the structure of the file, so only authorized personnel can see it and digital watermarking is used by companies so algorithms can find copywrited content, or a way for the original author to prove that they are responsible for the work in question.

Task 2

a) The information hidden are pictures in jpeg format, and it is hidden by merging it with another picture.
b)By merging the files it is possible to send from zip, files to jpeg files
C) HxD is a popular hexadecimal editor[1] in which was used to find the hidden message in the file “secret.bmp”:
There is much to see on but the importance in by looking at the decoded text, in there lies the way to solve this puzzle: because the usual way data is added by stenography is by performing the LSB method (Least significant bit) and that barely affects the pictures colours but it does change the bit composition and hence the files decrypted text, that is why its so important, by just looking at the decrypted text anything that looks different and also irregular becomes a suspicious area to look for instance here :

BM60 6 (@ @ ☺ ↑ 0 —♫ —♫
■ ■■■■ ■ ■ ■ ■ ■■ ■■ ■ ■■ ■■ ■■■■ ■ ■ ■■■ ■ ■ ■ ■ onn ☺☺
☺ ☺☺☺ ☺ ☺☺ ☺ ☺ ☺☺ ☺☺ττμ ■ ■■ ■■ ■■ ■■■■ ■■■
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

There are too many characters that change irregularly and makes it the most suspicious part :

Now if the LSB method is used then either the last or last 2 bits are arranged. And there it comes the long part I first tried to just pick the first bit and by rule first 0 then 1 and by doing that I got the following binary numbers:
01000011
01101111
01101110
01100111
01110010
01100001
01110100
01110101
01101100
01100001
01110100
01101001
01101111
01101110
01110011

Which when decoded it presents the following = “Congratulations”

And that should be it.

Task 3

a) "RK ERKT EHURMXD"= in vino veritas
b) It would be possible to decrypt in a short amount of time, the only issue is by having different keys, there would be many results to pick from so the person who has decrypted it has to pick from multiple possibilities, considering that the message is in Latin and quite unexpected, if the attacker only assumed that the English language was used then It might create its own sentence rather than the original word and ultimately be successful in hiding the message.

C) it is quite an issue to decrypt the message due to the following limitations:
Small amount of words; frequency table is not a valid method and the original message might be from another language (I would guess Latin maybe)

But with those limitations aside the following things can be said:
; "QMJ BPZ B XPJZ RZWJPAXQ LAD".

"B" is alone and starts a 3 letter word, It is probable that "B" is a vocal and assuming the message in in English then it would be I or A

APZ(could be act, aid, ask) since the word is behind an A I'm assuming that the word is a verb and these two look plausible aid a, ask a

P = I
Z = D

['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']

['', 'A', '', '', '', '', '', '', 'I', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', 'D']

XIJD = (bird, kind,)
(Bird), (kind)
X = B X=K
J = R J = N
QMJ AID A BIRD RDWRIABQ LAD

RZWJPAXQ
RDWRIABQ OR RDWNIAKQ
And this is were I can go because Im making so many assumptions that makes it difficult to keep guessing.

Task 4

The files contained will encrypt and decrypt any txt file, the manner I used to encrypt is by:

Using the Caesar cypher to change the words to in my case 7 shifts, A ---> H and then make a transposition method that has as a key 4213 and move those letters at that pace and everything in the text is affected by that key.

And in order to decrypt it,
For transposition method:
I separate the words by the inverse of the key
If the code was
4
2
1
3
It changes to
3
2
4
1
And for the Caesar cypher: the same code but the inverted is used H -----> A

Further details in both python files: “encryption_program.py” and “decryption_program.py”

Task 5

Key for the Caesar cypher = ['H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'A', 'B', 'C', 'D', 'E', 'F', 'G']
Key for the transposition method = [4,2,1,3]
File name= Fredric_Eriksson_Sepulveda.txt

Task 6

All work is in the python file “decode_techniques.py”
After the encrypted text was sent, I picked a work at random and chose “Albin_Johnsson_Ciphertext.txt”

I made a python file called decode techniques where I used the text file, what I did at the beginning was to create a frequency graph in order to find the most common letters in his text and I assumed that V = E and it worked got the whole text and apparently he didnt do any transposition methods.

Its a videogame dialogue.

So I decided to look at someone elses as well

“Chen Ningrui.txt”

Again frequency graph is used and In this file the most common letters are: M, Z, A, K, I, W, V, B, Q

And I assumed that M == E

And well that is all in the file...

Its a news story from the imprisonment of a druglord

Task 7

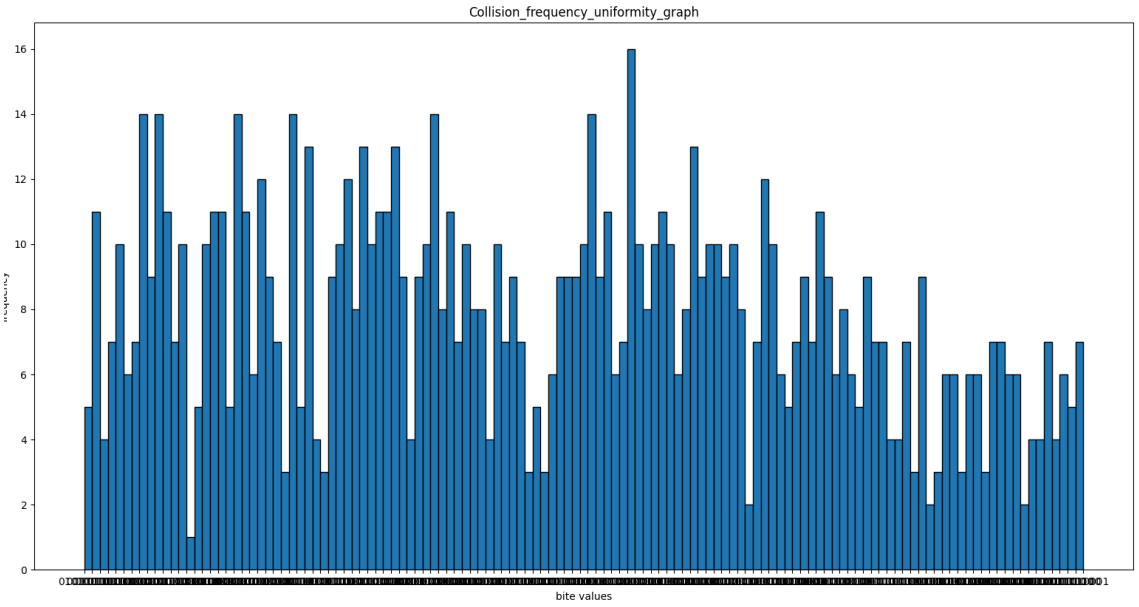
A) My hash algorithm is difficult to explain on paper but the idea is the following: all ASCII values in a text are added with each but the upcoming values are divided by len(of the string) and the value that divides increases the larger the text is, then all is rounded up

If the values are over 100 then that value is multiplied by 10^4 if its lower then 10^5, (the value cannot be under 10)

Then that value is added by all ASCII codes by the power of len(text) * 3

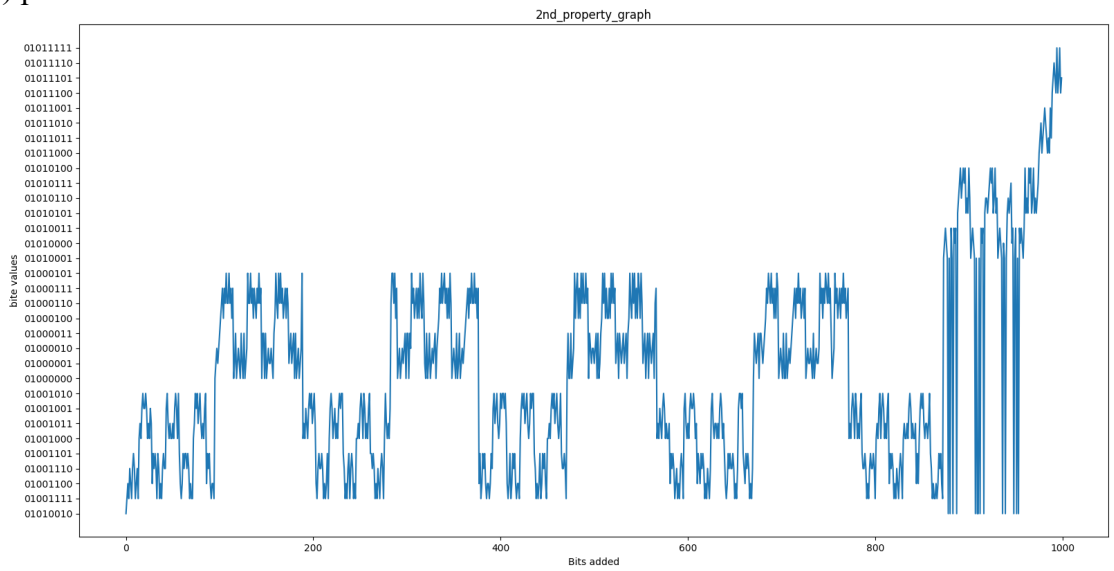
Once the value is determined the bites are made the following way: if the value is pair the value in the bite will be 0 if its not pair then its 1

B) part 1:



As it can be seen in the graph the amount of collisions in a span of 1000 tries it yields at least 100 collisions (all made by random text, it can be seen in the python code)

B) part 2:



As it can be seen in the graph after adding 1 bit the values fluctuate

C) the difference is that the secure hash function is made to be random and prevent all sorts of collisions the other method it just tries its best to avoid collisions

The first graph proves that is not a secure collision due to having at least a 10% margin of error in which in a secure hash function that margin is the lowest as possible and 10% is unacceptable , but the second graph it shows that it is at least a normal hash function.

Bibliography

[1] C. P. Fleeger, “Chapter 2 Toolbox,” in *Security in Computing FIFTH EDITION*, Pearson Education, 2015, pp. 120–125.

[2] M. Hörz, “HxD - Freeware Hex Editor and Disk Editor,” *mh*, 2003. [Online]. Available: https://mh-nexus.de/en/hxd/. [Accessed: 21-Nov-2020].

[3]