Linnéuniversitetet
Kalmar Växjö

Linnaeus University

1DV700 - Computer Security
Assignment 1

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

Setup Premises

Hex reader, HxD
Windows 10 OS
Python; Visual Studio Code

Task 1

a)
Symmetric encryption is “a method of cryptography where a single key is responsible for encrypting and decrypting data. The involved parties share that key, password, or passphrase, and they can use it to decrypt or encrypt any messages they want”.[1]
Asymmetric encryption is “a process that uses a pair of related keys -- one public key and one private key -- to encrypt and decrypt a message and protect it from unauthorized access or use. A public key is a cryptographic key that can be used by any person to encrypt a message so that it can only be deciphered by the intended recipient with their private key. A private key is shared only with the key's initiator”[2].

The differences between both encryption methods are: that Symmetric encryption only uses one key, and Asymmetric encryption uses two keys; one public key and one private key, so both methods could encrypt the message.

Encryption algorithms is “a component for electronic data transport security. Actual mathematical steps are taken and enlisted when developing algorithms for encryption purposes, and varying block ciphers are used to encrypt electronic data or numbers.”[3]
Hash algorithms is”a hash algorithm is a function that converts a data string into a numeric string output of fixed length. The output string is generally much smaller than the original data. Hash algorithms are designed to be collision-resistant, meaning that there is a very low probability that the same string would be created for different data.”[4]

The differences between both security methods are: Encryption algorithms operate by changing the contents of the file and creating an unintelligible document that could only be decrypted by those who have the decryption program and a key. Hash algorithms do not change the contents of the file, instead it creates a special fingerprint by considering everything written in the file and with that fingerprint it can be detected if the file is from the original author.

Compression is “the process of reducing the number of bits or bytes needed to represent a given set of data. It allows saving more data.”[5]
Hashing is ”is an algorithm that takes an arbitrary amount of data input—a credential—and produces a fixed-size output of enciphered text called a hash value, or just “hash.” That enciphered text can then be stored instead of the password itself, and later used to verify the user.”[6]

The difference between both processes is that the compression function is just to reduce the amount of bits a file uses, not to protect the contents of the file, hashing creates an enciphered text by using the contents of the file, then it can be used to identify the validity of said file.

b)
Steganography is a method to hide messages through images, one of the common steganography techniques used is: “to embed a text file into an image file. Anyone viewing the image file would see no difference between the original file and the file with the message embedded into it. This is accomplished by storing the message using least significant bits in the data file”[7]

Encryption is “the method by which information is converted into secret code that hides the information's true meaning. The science of encrypting and decrypting information is called *cryptography*.”[8]

Digital Watermarking is “the method of embedding data into digital multimedia content. This is used to verify the credibility of the content or to recognize the identity of the digital content's owner.”[9]

The difference between the methods are: Stenography is a method to send hidden messages, The most common method is to send hidden messages through picture files. Encryption is a method to hide the original contents of files and does so by changing the contents of the file so only authorized users can see it (those with a key). Digital watermarking is a method that Authors or companies use to identify their original content and prove that it is theirs by adding a hidden file or image inside their content.

Task 2

a)
Type: Images are hidden inside images.

How: Each channel (red, green, blue) of each pixel in an image is represented by an 8-bit value. To hide the secret image inside the cover image, we replace the **n** least significant bits of the cover pixel value with the same number of most significant bits from the secret pixel value[10]

Limitations: the higher amount of bits is used the higher the quality of the hidden image, the issue is that it becomes harder to hide.
b) It is possible to send text by hiding it under the hexadecimal code, since it would not make any perceivable changes to the image and only those who already know that the image has been tampered with will be able to find the hidden message.
c)
HxD is a popular hexadecimal editor in which was used to find the hidden message in the file "secret.bmp":
There is much to see on but the importance lies in the decoded text, in there is the solution to 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 original picture's colours but it does change the bit composition and hence the files decrypted text, that is why it's 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 suspicious:

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 when looking for letters, but it is not a rule to always follow and by doing that I got the following binary numbers:

1000011 C
01101111 o
01101110 n
01100111 g
01110010 r
01100001 a
01110100 t
01110101 u
01101100 l
01100001 a
01110100 t
01101001 i

01101111 o
01101110 n
01110011 s
00100001 !
11111111 (nothing else)

Which when decoded it presents the following = “Congratulations!”

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)

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 is in English then it would be I or A

APZ(could be act, aid, ask) since the word is behind an A I am 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 where I can go, because I am making so many assumptions that makes it difficult to keep guessing.

Task 4

The file “encrypt.py” can encrypt any txt file. The encryption methods are substitution and transposition

For the substitution method it is asked the user for a password, and from that password the key for the cypher is generated. The method used is to obtain all ASCII numbers in that password, add them together and then use the value%256 in order to obtain an 8 bit value in which would be used as a key for the Caesar Cypher.

For example if the password would be “password” the total ASCII value would be: 883 and then by using %256 the key value would be 115. After obtaining the key, A Caesar Cypher is used and the key determines the amount of shifts to the right.

The method of decrypting this is by asking the user again for the password and from that move back the amount of shifts used when the file was encrypted. The only issue is That I cannot decrypt the message in the right uppercase or lowercase order, So the final file will be all in Uppercase letters.

For the Transposition method a “pin” is asked from the user, and it should be 4 numbers and those numbers should not repeat otherwise errors may happen. from that pin a 0123 pattern is formed by judging the highest numbers in said pin code.

For example if the pin code is “5721” the pattern created would be 2310 and that key would determine how many spaces a set of four letters would shuffle, for example “abcd” will turn into “cdba”

In order to shuffle the letters It can be seen in the program that I form a new sentence by selecting the first 4 letters and the pin selects the order, as a method to not have out of index errors the last letters are counted and that determines how many letters would be shuffled.

In the decryption method was rather a difficult task to accomplish due to the fact that there was not a simple solution at the beginning due to, the fact that I could not find any patterns, So after some random attempts I stumbled upon the answer: which is by knowing the pin the order of the letters can be determined after some if conditions, It is not the most elegant but it works as intended.
Also the last letters follow a fixed pattern due to being hard to randomize and also probably helps in being more random and being harder to decipher.

How to use the program

When starting it it asks the user between encrypt and decrypt and the user selects it by pressing “1” or “2”, later the program asks for the txt file in which should be in the same folder as the program and the User has to input the whole name of the program and with .txt at the end, later the User has to decide whether to use A substitution encryption/decryption method or a Transposition method. after the instructions are completed a message saying “Encrypted” or “Done” will be shown.

Note: If any issues show up by selecting the txt file please edit the “read” string so it aligns with to where the txt file is located, The program assumes that the txt file is in the same directory.

Task 5

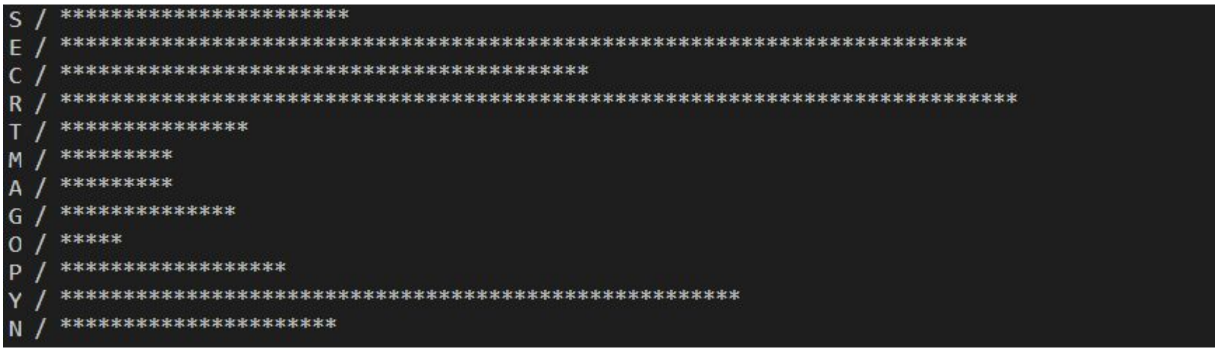
The File uploaded is called Fredric_Eriksson.txt

It follows my Transposition method which is found in the file “encrypt.py” and the pin used is “7691”

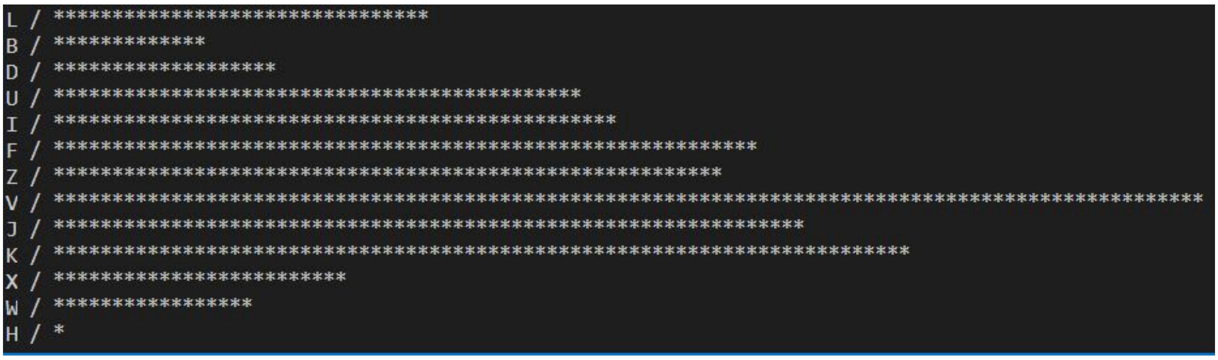
Task 6

The analysed text is from “Albin_Johnsson_Ciphertext.txt”

What I did at the beginning was to create a frequency graph in order to find the most common letters in his text:



```
S / ***********************
E / ************************************************************************
C / ******************************************
R / ****************************************************************************
T / ***************
M / *********
A / *********
G / **************
O / *****
P / ******************
Y / ******************************************************
N / **********************
```

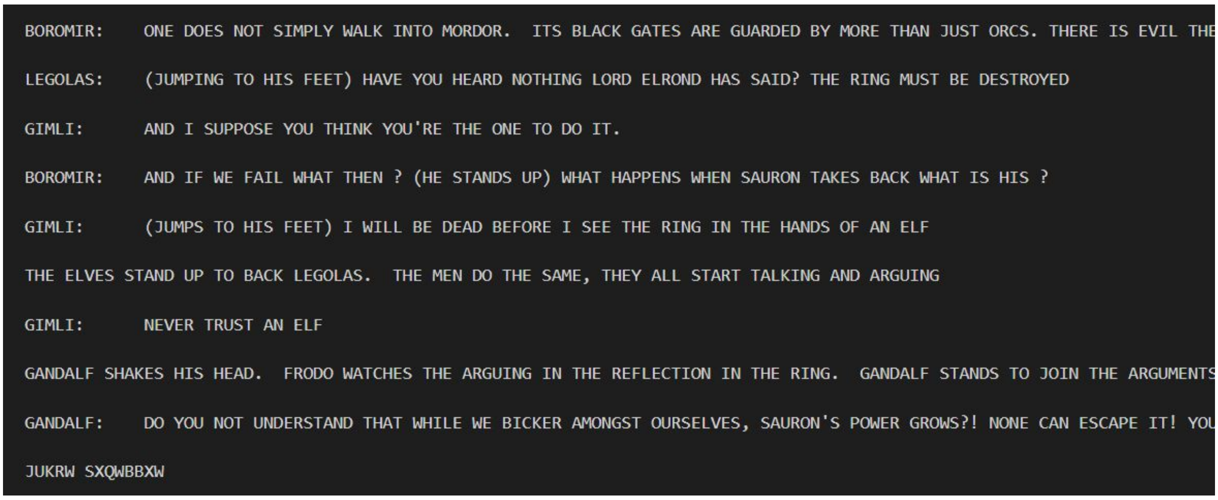


```
L / ********************************
B / *************
D / *******************
U / *********************************************
I / ************************************************
F / ************************************************************
Z / *********************************************************
V / **************************************************************************************************
J / ****************************************************************
K / *************************************************************************
X / *************************
W / *****************
H / *
```

And I assumed that V = E, by just an instinct I had from seeing the most used letter in the english alphabet tends to be E.

So I shifted the letters leftward by the number 17.

And I found the answer



```
BOROMIR:    ONE DOES NOT SIMPLY WALK INTO MORDOR.  ITS BLACK GATES ARE GUARDED BY MORE THAN JUST ORCS. THERE IS EVIL THE

LEGOLAS:    (JUMPING TO HIS FEET) HAVE YOU HEARD NOTHING LORD ELROND HAS SAID? THE RING MUST BE DESTROYED

GIMLI:      AND I SUPPOSE YOU THINK YOU'RE THE ONE TO DO IT.

BOROMIR:    AND IF WE FAIL WHAT THEN ? (HE STANDS UP) WHAT HAPPENS WHEN SAURON TAKES BACK WHAT IS HIS ?

GIMLI:      (JUMPS TO HIS FEET) I WILL BE DEAD BEFORE I SEE THE RING IN THE HANDS OF AN ELF

THE ELVES STAND UP TO BACK LEGOLAS.  THE MEN DO THE SAME, THEY ALL START TALKING AND ARGUING

GIMLI:      NEVER TRUST AN ELF

GANDALF SHAKES HIS HEAD.  FRODO WATCHES THE ARGUING IN THE REFLECTION IN THE RING.  GANDALF STANDS TO JOIN THE ARGUMENTS

GANDALF:    DO YOU NOT UNDERSTAND THAT WHILE WE BICKER AMONGST OURSELVES, SAURON'S POWER GROWS?! NONE CAN ESCAPE IT! YOU

JUKRW SXQWBBXW
```

the name at the end was added after he encrypted his text.

Task 7

a) This hash function wants to find a key by means of counting all ASCII values in a text and then %256 the value obtained in order to generate the key.

The file name is "Simple_hash_function.py" when starting the program it will ask you for the name of the txt file (make sure it is located in the same directory)

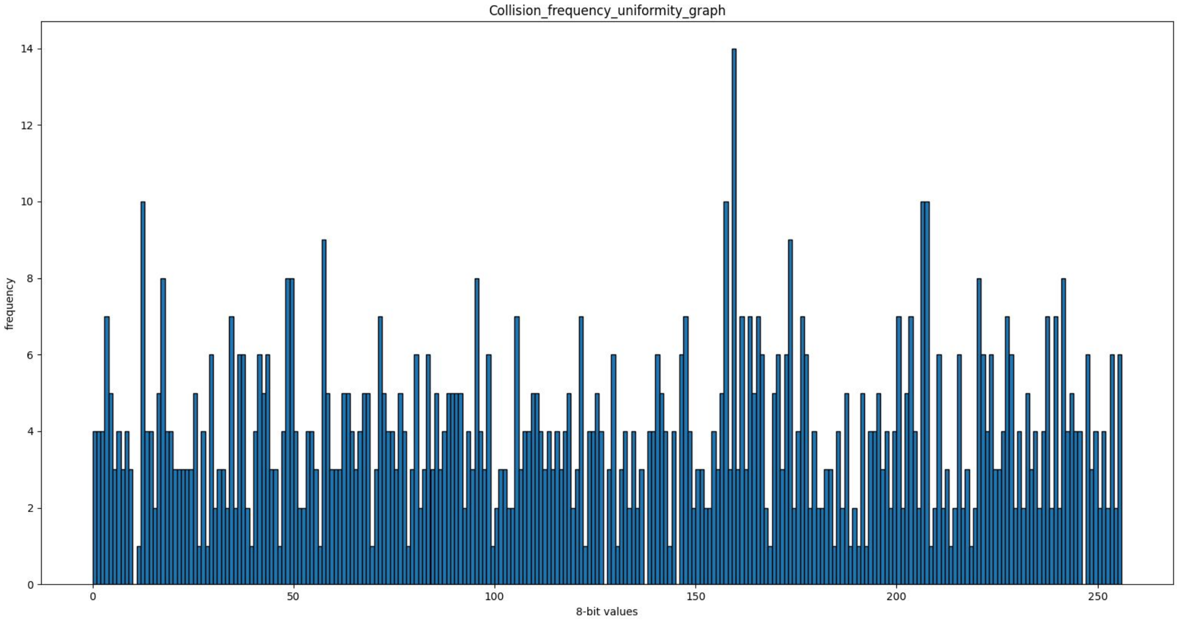
and in the bottom it will print the hash value for the txt file

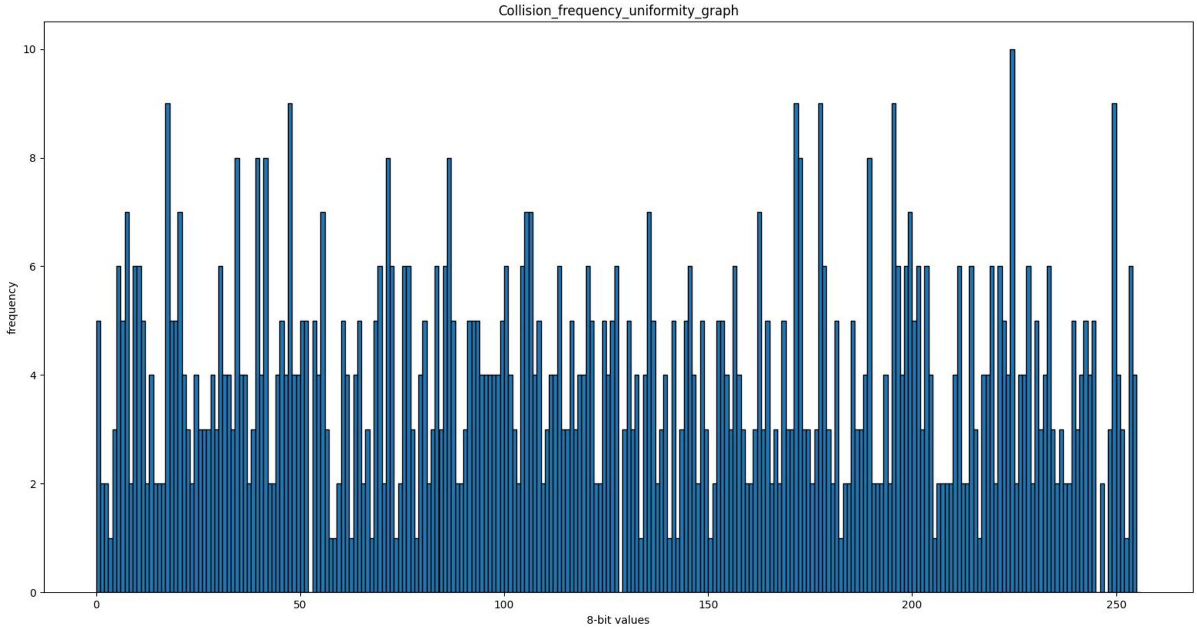
If you are interested in how the following graphs were made I also left the procedures in the file it will be under the comment "Task 4 B)" feel free to ignore it if was not part of the procedure

B)

Uniformity:

after 1000 tests these are some of the results





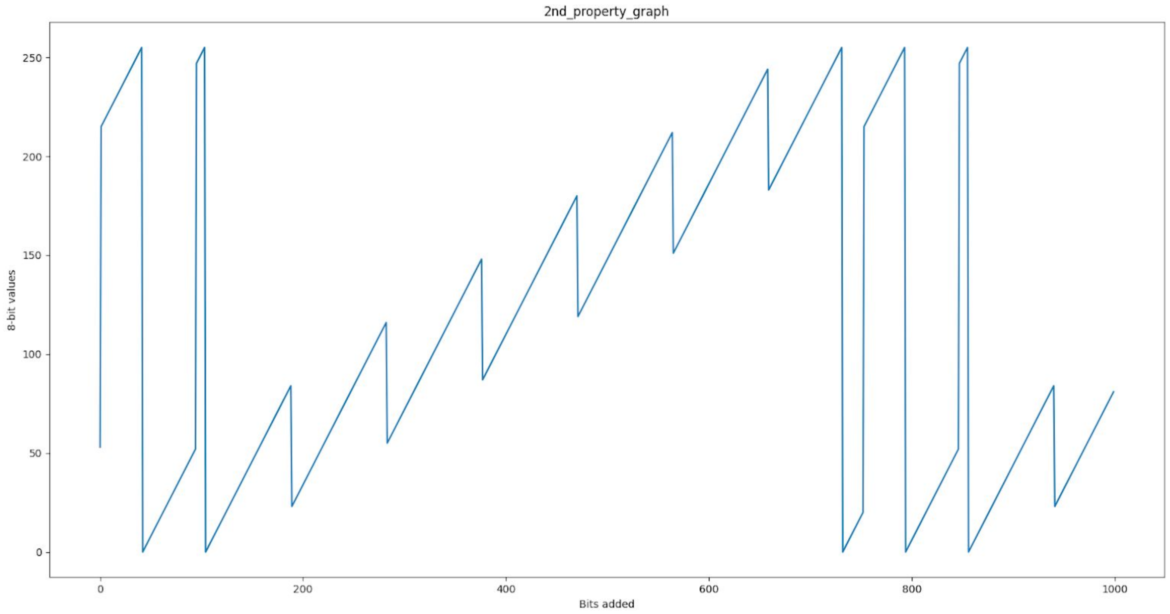
The maximum frequency seen was 14. so at maximum there could be 14 collisions with 1 value.

In order to make this graph I used the random module in order to determine the length of the text, and later make random letters by turning the ASCII values into letters that are also randomly generated.

The method utilized is rather simplistic due to me not using actual words but rather random letters strung together such as for example: “j ,Krq1”, In that manner this method might not follow standards but I ultimately think that it does not matter because the focus is in the hash value and it delivers.

2nd Property:

The values sometimes have a linear relationship, but then at certain points it changes dramatically, then again it follows a linear relationship and repeats.



I made this graph by adding “AAA” into the text then replacing the last A to B and it keeps going until the second A changes to B and repeats until the counter reaches a thousand.

The method employed is rather sophisticated I might suggest so the method utilized should not have any issues regarding the results.

C)

The difference between both are that normal hash functions have the only aim of making a hash value it has no other aim, in the other hand Secure hash functions in the addition of making hash values it also has the aim of minimizing collisions, which means that it wants to make as many unique hash values as possible when different texts are introduced.

My hash function is not secure by just examining both graphs, in the uniformity graph for it to be considered secure then the maximum frequency should be 1 not 14 for 1000 different x values.

In the second graph: a pattern can be found and be easily exploitable, for it to be secure then it should look random and no patterns could be made.

‘

Bibliography

[1] D. Bisson, "What Is Symmetric Encryption | Venafi", *Venafi.com*, 2020. [Online]. Available: https://www.venafi.com/blog/what-symmetric-encryption. [Accessed: 04- Dec- 2020].

[2]M. Rouse, "What is Asymmetric Cryptography and How Does it Work?", *SearchSecurity*, 2020. [Online]. Available: https://searchsecurity.techtarget.com/definition/asymmetric-cryptography. [Accessed: 04- Dec- 2020].

[3]"What is an Encryption Algorithm? - Definition from Techopedia", *Techopedia.com*, 2020. [Online]. Available: https://www.techopedia.com/definition/1778/encryption-algorithm#:~:text=An%20encryption%20algorithm%20is%20a,encrypt%20electronic%20data%20or%20numbers. [Accessed: 06- Dec- 2020].

[4]"Hash algorithm - Glossary - Federal Agencies Digitization Guidelines Initiative", *Digitizationguidelines.gov*, 2020. [Online]. Available: http://www.digitizationguidelines.gov/term.php?term=hashalgorithm. [Accessed: 06- Dec- 2020].

[5]R. Sharma and S. Bollavarapu, "Data Security using Compression and Cryptography Techniques", *Semanticscholar.org*, 2020. [Online]. Available: https://www.semanticscholar.org/paper/Data-Security-using-Compression-and-Cryptography-Sharma-Bollavarapu/0b577919bdd5e0767c733cfbe94897d3890aaede. [Accessed: 06- Dec- 2020].

[6]"What are cryptographic hash functions? | Synopsys", *Software Integrity Blog*, 2020. [Online]. Available: https://www.synopsys.com/blogs/software-security/cryptographic-hash-functions/#:~:text=A%20cryptographic%20hash%20function%20is,used%20to%20verify%20the%20user. [Accessed: 06- Dec- 2020].

[7]How does steganography work and does it threaten enterprise data? PDF file. [Accessed: 06- Dec- 2020].

[8]M. Rouse, "What is Encryption and How Does it Work?", SearchSecurity, 2020. [Online]. Available: https://searchsecurity.techtarget.com/definition/encryption#:~:text=Encryption%20is%20the%20method%20by,encrypted%20data%20is%20called%20ciphertext. [Accessed: 06- Dec- 2020].

[9]"What is Digital Watermarking? - Definition from Techopedia", *Techopedia.com*, 2020. [Online]. Available: https://www.techopedia.com/definition/24927/digital-watermarking. [Accessed: 06- Dec- 2020].