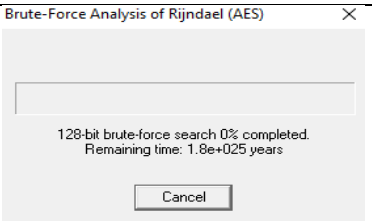
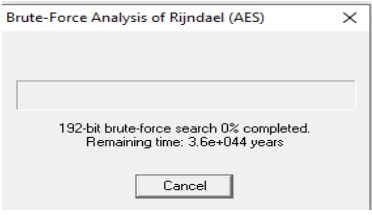
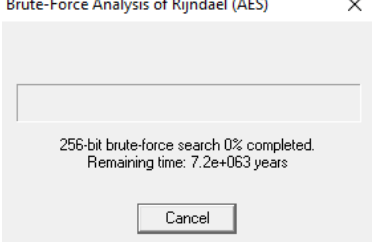
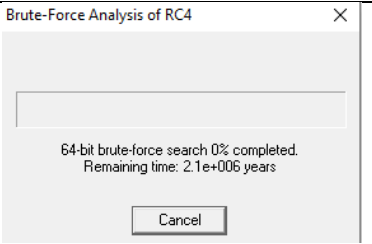


## I. Tasks:

1. Evaluate the time needed to find a full key of 64, 128, 192, 256 bits.
2. Compare the time it takes to find keys of the same length (e.g. 128 bits) for different algorithms.

= > Results at the below tables are related to both task1 and task2. Observations related to analysis will be explained at the end.

ALGORITHMS	BITS	TIME NEEDED
AES	128 bits	
	192 bits	
	256 bits	
RC4	64 bits	

RC4	128 bits	<div>Brute-Force Analysis of RC4</div> <div></div> <div>128-bit brute-force search 0% completed. Remaining time: 4.9e+025 years</div> <div>Cancel</div>
DES(ECB)	64 bits	<div>Brute-Force Analysis of DES (ECB)</div> <div></div> <div>56-bit brute-force search 0% completed. Remaining time: 1.9e+004 years</div> <div>Cancel</div>
	128 bits	<div>Brute-Force Analysis of Triple DES (ECB)</div> <div></div> <div>112-bit brute-force search 0% completed. Remaining time: 2.4e+021 years</div> <div>Cancel</div>
TRIPLE DES(ECB)	128 bits	<div>Brute-Force Analysis of Triple DES (ECB)</div> <div></div> <div>112-bit brute-force search 0% completed. Remaining time: 2.5e+021 years</div> <div>Cancel</div>
IDEA	128 bits	<div>Brute-Force Analysis of IDEA</div> <div></div> <div>128-bit brute-force search 0% completed. Remaining time: 1.4e+026 years</div> <div>Cancel</div>

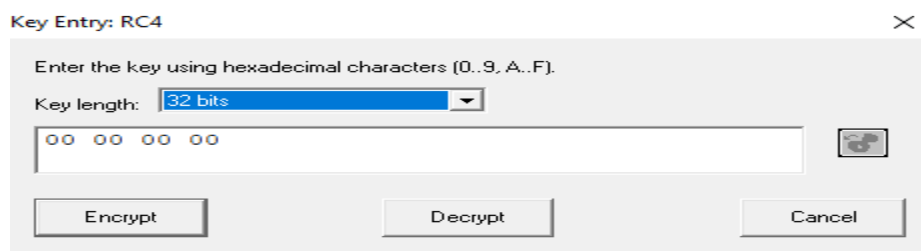
## Observations:

**Task 1 = >** According to analysis observed in the above tables, since the size of bits are increased, Brute Force attacking will be taken that time related. So, attempting every combination one by one becomes the disadvantages of the Brute Force that makes the decrypt harder and longer.

**Task 2 = >** According to analysis observed in the above tables, when we comparing the difference algorithms with same bits key size seems not a main thing. I guess depends on how is algorithms fast to encrypt.

### 3. Determine the time needed to find a key if the unknown number of bits of the key is: 4, 8, 12, 16, 20, 24, 30, 34. (One star in the key equals 4 bits).

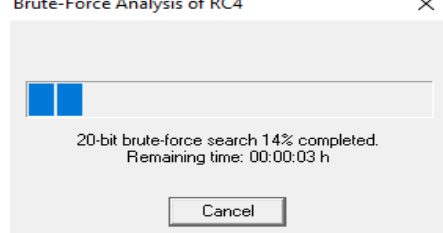
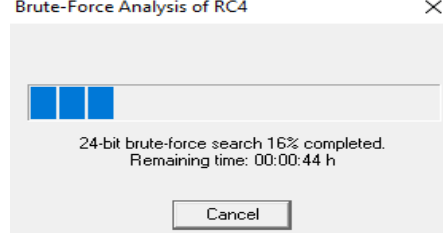
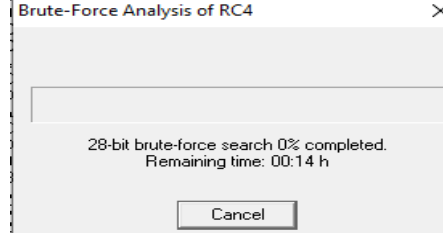
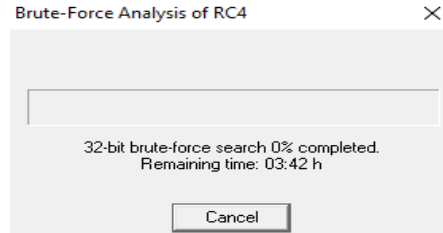
= > For the RC4 algorithm when encrypting the data I keep the key length 32 bits and 00000 to numbers before the Checking time remaining for Brute Force.

A screenshot of a 'Key Entry: RC4' dialog box. The title bar says 'Key Entry: RC4' with a close button. The main text says 'Enter the key using hexadecimal characters (0..9, A..F)'. Below this is a 'Key length:' label followed by a dropdown menu showing '32 bits'. Underneath is a text input field containing '00 00 00 00' with a small icon to its right. At the bottom are three buttons: 'Encrypt', 'Decrypt', and 'Cancel'.

= > Since the every star in the key is equals to 4 bits, in order to check the remaining time for different bits, i deleted 1 star starting with the 32 bits. So I observed the remaining time difference between different bits . According to results since the bits increased in the Brute Forcing time remaining will be simultaneously increased.

-> Analysis for the time remaining section until 20 bits wasn't possible to observed because they were less than 1 second.

Bits	Remaining Time
4 bits	1 second or less
8 bits	1 second or less
12 bits	1 second or less
16 bits	1 second or less

20 bits	00:00:03h	 <p>20-bit brute-force search 14% completed. Remaining time: 00:00:03 h</p> <p>Cancel</p>
24 bits	00:00:46h	 <p>24-bit brute-force search 16% completed. Remaining time: 00:00:44 h</p> <p>Cancel</p>
28 bits	00:14h	 <p>28-bit brute-force search 0% completed. Remaining time: 00:14 h</p> <p>Cancel</p>
32 bits	03:44h	 <p>32-bit brute-force search 0% completed. Remaining time: 03:42 h</p> <p>Cancel</p>

#### 4. Check whether the position of the unknown bits in the key affects the key search time.

= > According to my observations, position of the unknown bits in the key doesn't really affects the key search time. The thing affects the key search time is an size of the bits. Search time simultaneously increasing with the size of bits. I didn't observed any changes even in the remaining time when changing the numbers(00 00) with the randomly numbered. Not sure.

#### 5) Evaluate the performance of the cryptanalysis tools.

= > Since the cryptanalysis is the process of studying cryptographic system to look for weakness or leaks of information, I am not always be able to find the requested analysis type or sth. shared in the exercise pdf easily. Because of limited information about the app. But generally its friendly, well-documented system with its interface that I am able to find what I'm looking in a couple of minutes. The only suggestion I have for the tools we're using is to utilize different methods to find the key, but that's what I expected because algorithms change, thus we need to indicate which encryption method we want to use.

### **5.1 Do we always get the correct key?**

= > I say no. Because some of the algorithms are hard and long to decrypt, it doesn't allow us to find the key. Some of time remaining are like years so even though it depends on your computer it's not in every situation.

### **5.2 Does the number of the bits search affect the quality of the key being restored?**

= > In my opinion number of the bits doesn't affects the quality of the key, as I saw from the experiences observed at the above tables, capacity of the bits just increases the time remaining.

### **5.3 Does the position of unknown bits affect the quality of the key being restored?**

= > In my opinion position of the unknown bits doesn't affect the quality of the key. Not sure.

## **II. Questions:**

### **1. Can modern block algorithms be considered safe (referring to the above experiments)?**

= > According with my search although it is highly efficient in 128-bit form, some algorithms like AES uses keys of 192 and 256 bits for heavy-duty encryption purposes. So with the longer key we are able to provide safety to data.

### **2. What length of the key offers us a sufficient level of security for particular algorithms? Why?**

= > The longer the the secret key , the harder it is for attacker to guess via Brute Force attack. However, AES-256 is not just twice as strong as AES-128. For 128 and 256 bits there are  $2^{128}$ ,  $2^{256}$  potential secret keys.

But eventhough 256 bits is much more safety than 128 it takes so much time to decrypt, 128-bits key is secure against attack by modern technology.

### **3. Does the size of the ciphertext influence the possibility of breaking it?**

= > Yes, even letters influence. A single bit error in a ciphertext blocks affects the decryption of all subsequent blocks. Hiding one letter in a text will be easier than a long text.

### **4. Does the format and earlier processing of the document affect the possibility of its cryptanalysis (e.g. compression etc.)?**

= > Yes format and earlier processing of the document affect the possibility of its cryptanalysis. Earlier processing affects the cryptanalysis and difference formats have different entropy and complexity.

**5. How many possible passwords can we check in a year of continuous work on one computer, which checks one million passwords per second ( 20 2 )?**

= > Speed depending on password strength: computer programs used for brute force attacks can check anywhere from 10.000 to 1 billion passwords per second. There are 94 numbers, letters, and symbols on a standard keyboard. In total, they can generate around two hundred billion 8-character password.

So lets make a calculation that checks possible password for one continuously year:

Lets say, Computer going to check 1.000.000 possible solution per second:

One second =>  $X = 1.000.000$

One minute = >  $x * 60$

One hour = >  $x * 60 * 60$

One day = >  $x * 60 * 60 * 24$

One year = >  $x * 60 * 60 * 24 * 365$

31.536.000.000.000

**6. What does this result say about the security of modern algorithms?**

= > Usually attacks are performed by GPUs. So the main parameters when cracking the data are GPU count and their power and Password algorithm. 1m per second to try possible password is impossible for home computers. An encryption key is a variable value that is applied using an algorithm to a string or block of unencrypted text to produce encrypted text or to decrypt encrypted text. The length of the key is a factor in considering how difficult it will be to decrypt the text in a given message.

**I. Tasks:**

